BESS Working Group
Internet Draft
Intended Status: Proposed Standard
Expires: April 26, 2016

P. Brissette
Cisco System
H. Shah
Ciena Corporation
Z. Li
Huawei Technologies
A. liu
Ericsson
K. Tiruveedhula
T. Singh
Juniper Networks
I. Hussain
Infinera Corporation
J. Rabadan
Alcatel-Lucent

October 16, 2015

Yang Data Model for EVPN
draft-brissette-bess-evpn-yang-00

Abstract

This document describes a YANG data model for Ethernet VPN services. The model is agnostic of the underlay. It apply to MPLS as well as to VxLAN encapsulation. The model is also agnostic of the services including E-LAN, E-LINE and E-TREE services. The merging of this model with L2 services model is for future investigation. Any "add-on" features such as EVPN IRB, EVPN overlay, etc. are for future investigation. This document mainly focuses on EVPN instance framework.

Status of this Memo

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

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

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

Copyright and License Notice

Convention

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

Table of Contents

1. Introduction 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 services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The EVPN basic framework definition is covered first. Merging with L2 services model is left for future study. The EVPN basic framework consist of two modules: evpn and ethernet-segment. These models completely orthogonal. They usually work in pair but user can definitely use one or the other for its own need.

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

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

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

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

The ethernet-segment data object model defined in this document refer to specific an interface. The interface can be a physical interface,

a bundle interface or virtual interface. The latter includes pseudowires. The purpose of creating a separate module is due to the fact that it can be used without having the need to have evpn configured as layer 2 service. For example, an access node can be dual-homed to two service nodes servicing a VPLS core. The access connectivity can be represented by an ethernet-segment where EVPN BGP DF election is performed over both service nodes. The core remains VPLS. Therefore, there is no EVPN instance being used here.

2. Specification of Requirements

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

3. EVPN YANG Model

3.1. Overview

Two top level module, ethernet-segment and evpn, are defined. The ethernet-segment contains a list of interface to which any ethernet-segment attributes are configured/applied.

The evpn module has 2 main containers: common and instance. The first one has common attributes to all VPNs where as the latter has attributes specific to an EVI. This document state the scope of the EVPN object models definition. The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

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

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

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

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

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

3.2 Ethernet-Segment Model

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

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

3.3 EVPN Model

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

```
module: ietf-evpn
   +--rw evpn
```

```
         +--rw common
         |  +--rw (replication-type)?
         |     +--:(ingress-replication)
         |     |  +--rw ingress-replication?   boolean
         |     +--:(p2mp-replication)
         |        +--rw p2mp-replication?      boolean
         +--rw evpn-instances
            +--rw evpn-instance* [name]
               +--rw name              string
               +--rw evi?              uint32
               +--rw source-bmac?      yang:hex-string
               +--rw evpn-arp-proxy?   boolean
               +--rw nd-arp-proxy?     boolean
               +--rw bgp-parameters
                  +--rw common
                     +--rw route-distinguisher?   string
                     +--rw vpn-targets* [rt-value]
                        +--rw rt-value    string
                        +--rw rt-type     bgp-rt-type
```

4. YANG Module

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

4.1 Ethernet Segment Yang Module

```
<CODE BEGINS>
module ietf-ethernet-segment {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ethernet-segment";
  prefix "es";

  import ietf-evpn {
    prefix "evpn";
  }

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

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

  /* EVPN Ethernet Segment YANG Model */

  container ethernet-segments {
```

```
       description "ethernet-segment";
       list ethernet-segment {
         key "name";
         leaf name {
           type string;
           description "Name of the ethernet segment";
         }
         leaf esi {
           type empty;
           description "esi";
         }
         choice active-mode {
           mandatory true;
           description "Choice of active mode";
           case single-active {
             leaf single-active-mode {
               type empty;
               description "single-active-mode";
             }
           }
           case all-active {
             leaf all-active-mode {
               type empty;
               description "all-active-mode";
             }
           }
         }
         uses evpn:bgp-parameters-grp;
         container df-election {
           description "df-election";
           choice df-election-method {
             description "Choice of df election method";
             case highest-random-weight {
               leaf enable-hrw {
                 type empty;
                 description "enable-hrw";
               }
             }
           }
           leaf election-wait-time {
             type uint32;
             description "election-wait-time";
           }
         }
         description "An ethernet segment";
       }
     }
   }
```

<CODE ENDS>

4.2 EVPN Yang Module

```
<CODE BEGINS>
module ietf-evpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-evpn";
  prefix "evpn";

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

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

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

  /* Typedefs */

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

  /* Groupings */

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

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

      /* EVPN YANG Model */

      container evpn {
        description "evpn";
        container common {
          description "common epn attributes";
          choice replication-type {
            description "A choice of replication type";
            case ingress-replication {
              leaf ingress-replication {
                type boolean;
                description "ingress-replication";
              }
            }
            case p2mp-replication {
              leaf p2mp-replication {
                type boolean;
                description "p2mp-replication";
              }
            }
          }
        }
        container evpn-instances {
          description "evpn-instances";
          list evpn-instance {
            key "name";
            description "An EVPN instance";
```

```
            leaf name {
              type string;
              description "Name of EVPN instance";
            }
            leaf evi {
              type uint32;
              description "evi";
            }
            leaf source-bmac {
              type yang:hex-string;
              description "source-bmac";
            }
            leaf evpn-arp-proxy {
              type boolean;
              description "evpn-arp-proxy";
            }
            leaf nd-arp-proxy {
              type boolean;
              description "nd-arp-proxy";
            }
            uses bgp-parameters-grp;
          }
        }
      }
    }
    <CODE ENDS>
```

5. Security Considerations

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

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

6. IANA Considerations

None.

7. Acknowledgments

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

8. References

8.1. Normative References

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

8.2. Informative References

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

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

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

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

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

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

Authors' Addresses

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

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

Zhenbin Li
Huawei Technologies
EMail: lizhenbin@huawei.com

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

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

Tapraj Singh
Juniper Networks
EMail: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

Jorge Rabadan
Alcatel-Lucent
EMail: jorge.rabadan@alcatel-lucent.com

L2VPN Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

A. Sajassi (Editor)
Cisco
J. Drake (Editor)
Juniper
Nabil Bitar
Verizon
Aldrin Isaac
Juniper
James Uttaro
AT&T
W. Henderickx
Alcatel-Lucent

Expires: April 19, 2016

October 19, 2015

A Network Virtualization Overlay Solution using EVPN
draft-ietf-bess-evpn-overlay-02

Abstract

This document describes how Ethernet VPN (EVPN) [RFC7432] can be used as an Network Virtualization Overlay (NVO) solution and explores the various tunnel encapsulation options over IP and their impact on the EVPN control-plane and procedures. In particular, the following encapsulation options are analyzed: VXLAN, NVGRE, and MPLS over GRE.

Table of Contents

1 Introduction 4
2 Specification of Requirements 5
3 Terminology 5
4 EVPN Features 6
5 Encapsulation Options for EVPN Overlays 7
5.1 VXLAN/NVGRE Encapsulation 7
5.1.1 Virtual Identifiers Scope 8
5.1.1.1 Data Center Interconnect with Gateway 8
5.1.1.2 Data Center Interconnect without Gateway 9
5.1.2 Virtual Identifiers to EVI Mapping 9
5.1.2.1 Auto Derivation of RT 10
5.1.3 Constructing EVPN BGP Routes 11
5.2 MPLS over GRE 13
6 EVPN with Multiple Data Plane Encapsulations 13
7 NVE Residing in Hypervisor 14
7.1 Impact on EVPN BGP Routes & Attributes for VXLAN/NVGRE Encapsulation 14
7.2 Impact on EVPN Procedures for VXLAN/NVGRE Encapsulation . . 15
8 NVE Residing in ToR Switch 15
8.1 EVPN Multi-Homing Features 16
8.1.1 Multi-homed Ethernet Segment Auto-Discovery 16
8.1.2 Fast Convergence and Mass Withdraw 16
8.1.3 Split-Horizon 16
8.1.4 Aliasing and Backup-Path 17
8.1.5 DF Election 17
8.2 Impact on EVPN BGP Routes & Attributes 18
8.3 Impact on EVPN Procedures 18
8.3.1 Split Horizon 19
8.3.2 Aliasing and Backup-Path 19

9 Support for Multicast 19
10 Data Center Interconnections - DCI 20
10.1 DCI using GWs 20
10.2 DCI using ASBRs 21
10.2.1 ASBR Functionality with NVEs in Hypervisors 22
10.2.2 ASBR Functionality with NVEs in TORs 22
11 Acknowledgement 24
12 Security Considerations 24
13 IANA Considerations 25
14 References 25
14.1 Normative References 25
14.2 Informative References 25
Contributors 26
Authors' Addresses 26

1 Introduction

In the context of this document, a Network Virtualization Overlay (NVO) is a solution to address the requirements of a multi-tenant data center, especially one with virtualized hosts, e.g., Virtual Machines (VMs). The key requirements of such a solution, as described in [Problem-Statement], are:

- Isolation of network traffic per tenant

- Support for a large number of tenants (tens or hundreds of thousands)

- Extending L2 connectivity among different VMs belonging to a given tenant segment (subnet) across different PODs within a data center or between different data centers

- Allowing a given VM to move between different physical points of attachment within a given L2 segment

The underlay network for NVO solutions is assumed to provide IP connectivity between NVO endpoints (NVEs).

This document describes how Ethernet VPN (EVPN) can be used as an NVO solution and explores applicability of EVPN functions and procedures. In particular, it describes the various tunnel encapsulation options for EVPN over IP, and their impact on the EVPN control-plane and procedures for two main scenarios:

a) when the NVE resides in the hypervisor, and
b) when the NVE resides in a Top of Rack (ToR) device

Note that the use of EVPN as an NVO solution does not necessarily mandate that the BGP control-plane be running on the NVE. For such scenarios, it is still possible to leverage the EVPN solution by using XMPP, or alternative mechanisms, to extend the control-plane to the NVE as discussed in [L3VPN-ENDSYSTEMS].

The possible encapsulation options for EVPN overlays that are analyzed in this document are:

- VXLAN and NVGRE
- MPLS over GRE

Before getting into the description of the different encapsulation options for EVPN over IP, it is important to highlight the EVPN solution’s main features, how those features are currently supported,

and any impact that the encapsulation has on those features.

2 Specification of Requirements

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

3 Terminology

NVO: Network Virtualization Overlay

NVE: Network Virtualization Endpoint

VNI: Virtual Network Identifier (for VXLAN)

VSID: Virtual Subnet Identifier (for NVGRE)

EVPN: Ethernet VPN

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

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

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

Ethernet Segment Identifier (ESI): A unique non-zero identifier that identifies an Ethernet segment is called an 'Ethernet Segment Identifier'.

Ethernet Tag: An Ethernet tag identifies a particular broadcast domain, e.g., a VLAN. An EVPN instance consists of one or more broadcast domains.

PE: Provider Edge device.

Single-Active Redundancy Mode: When only a single PE, among all the PEs attached to an Ethernet segment, is allowed to forward traffic to/from that Ethernet segment for a given VLAN, then the Ethernet segment is defined to be operating in Single-Active redundancy mode.

All-Active Redundancy Mode: When all PEs attached to an Ethernet

segment are allowed to forward known unicast traffic to/from that Ethernet segment for a given VLAN, then the Ethernet segment is defined to be operating in All-Active redundancy mode.

4 EVPN Features

EVPN was originally designed to support the requirements detailed in [RFC7209] and therefore has the following attributes which directly address control plane scaling and ease of deployment issues.

1) Control plane traffic is distributed with BGP and Broadcast and Multicast traffic is sent using a shared multicast tree or with ingress replication.

2) Control plane learning is used for MAC (and IP) addresses instead of data plane learning. The latter requires the flooding of unknown unicast and ARP frames; whereas, the former does not require any flooding.

3) Route Reflector is used to reduce a full mesh of BGP sessions among PE devices to a single BGP session between a PE and the RR. Furthermore, RR hierarchy can be leveraged to scale the number of BGP routes on the RR.

4) Auto-discovery via BGP is used to discover PE devices participating in a given VPN, PE devices participating in a given redundancy group, tunnel encapsulation types, multicast tunnel type, multicast members, etc.

5) All-Active multihoming is used. This allows a given customer device (CE) to have multiple links to multiple PEs, and traffic to/from that CE fully utilizes all of these links. This set of links is termed an Ethernet Segment (ES).

6) When a link between a CE and a PE fails, the PEs for that EVI are notified of the failure via the withdrawal of a single EVPN route. This allows those PEs to remove the withdrawing PE as a next hop for every MAC address associated with the failed link. This is termed 'mass withdrawal'.

7) BGP route filtering and constrained route distribution are leveraged to ensure that the control plane traffic for a given EVI is only distributed to the PEs in that EVI.

8) When a 802.1Q interface is used between a CE and a PE, each of the VLAN ID (VID) on that interface can be mapped onto a bridge table (for upto 4094 such bridge tables). All these bridge tables may be

mapped onto a single MAC-VRF (in case of VLAN-aware bundle service).

9) VM Mobility mechanisms ensure that all PEs in a given EVI know the ES with which a given VM, as identified by its MAC and IP addresses, is currently associated.

10) Route Targets are used to allow the operator (or customer) to define a spectrum of logical network topologies including mesh, hub & spoke, and extranets (e.g., a VPN whose sites are owned by different enterprises), without the need for proprietary software or the aid of other virtual or physical devices.

11) Because the design goal for NVO is millions of instances per common physical infrastructure, the scaling properties of the control plane for NVO are extremely important. EVPN and the extensions described herein, are designed with this level of scalability in mind.

5 Encapsulation Options for EVPN Overlays

5.1 VXLAN/NVGRE Encapsulation

Both VXLAN and NVGRE are examples of technologies that provide a data plane encapsulation which is used to transport a packet over the common physical IP infrastructure between VXLAN Tunnel End Points (VTEPs) in VXLAN network and Network Virtualization Endpoints (NVEs) in NVGRE network. Both of these technologies include the identifier of the specific NVO instance, Virtual Network Identifier (VNI) in VXLAN and Virtual Subnet Identifier (VSID) in NVGRE, in each packet.

Note that a Provider Edge (PE) is equivalent to a VTEP/NVE.

VXLAN encapsulation is based on UDP, with an 8-byte header following the UDP header. VXLAN provides a 24-bit VNI, which typically provides a one-to-one mapping to the tenant VLAN ID, as described in [RFC7348]. In this scenario, the ingress VTEP does not include an inner VLAN tag on the encapsulated frame, and the egress VTEP discards the frames with an inner VLAN tag. This mode of operation in [RFC7348] maps to VLAN Based Service in [RFC7432], where a tenant VLAN ID gets mapped to an EVPN instance (EVI).

VXLAN also provides an option of including an inner VLAN tag in the encapsulated frame, if explicitly configured at the VTEP. This mode of operation can map to VLAN Bundle Service in [RFC7432] because all the tenant’s tagged frames map to a single bridge table / MAC-VRF, and the inner VLAN tag is not used for lookup by the disposition PE

when performing VXLAN decapsulation as described in section 6 of [RFC7348].

[NVGRE] encapsulation is based on [GRE] and it mandates the inclusion of the optional GRE Key field which carries the VSID. There is a one-to-one mapping between the VSID and the tenant VLAN ID, as described in [NVGRE] and the inclusion of an inner VLAN tag is prohibited. This mode of operation in [NVGRE] maps to VLAN Based Service in [RFC7432].

As described in the next section there is no change to the encoding of EVPN routes to support VXLAN or NVGRE encapsulation except for the use of BGP Encapsulation extended community. However, there is potential impact to the EVPN procedures depending on where the NVE is located (i.e., in hypervisor or TOR) and whether multi-homing capabilities are required.

5.1.1 Virtual Identifiers Scope

Although VNI or VSID are defined as 24-bit globally unique values, there are scenarios in which it is desirable to use a locally significant value for VNI or VSID, especially in the context of data center interconnect:

5.1.1.1 Data Center Interconnect with Gateway

In the case where NVEs in different data centers need to be interconnected, and the NVEs need to use VNIs or VSIDs as a globally unique identifiers within a data center, then a Gateway needs to be employed at the edge of the data center network. This is because the Gateway will provide the functionality of translating the VNI or VSID when crossing network boundaries, which may align with operator span of control boundaries. As an example, consider the network of Figure 1 below. Assume there are three network operators: one for each of the DC1, DC2 and WAN networks. The Gateways at the edge of the data centers are responsible for translating the VNIs / VSIDs between the values used in each of the data center networks and the values used in the WAN.

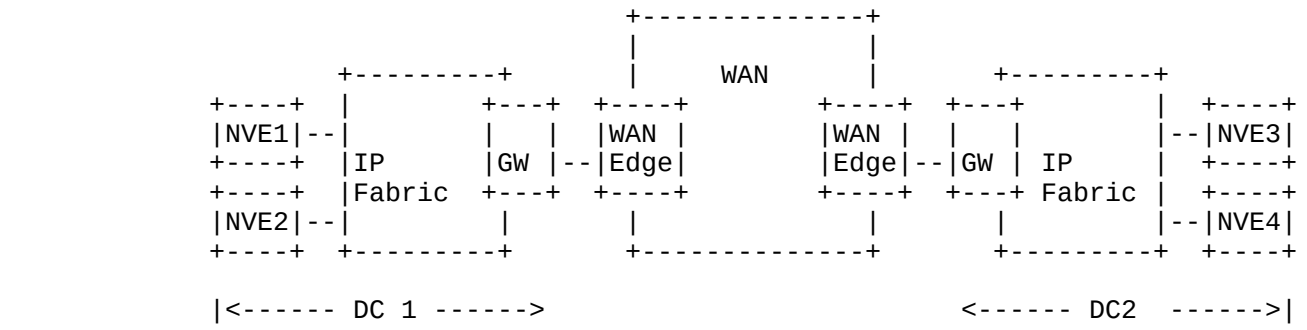


Figure 1: Data Center Interconnect with Gateway

5.1.1.2 Data Center Interconnect without Gateway

In the case where NVEs in different data centers need to be interconnected, and the NVEs need to use locally assigned VNIs or VSIDs (e.g., as MPLS labels), then there may be no need to employ Gateways at the edge of the data center network. More specifically, the VNI or VSID value that is used by the transmitting NVE is allocated by the NVE that is receiving the traffic (in other words, this is a "downstream assigned" MPLS label). This allows the VNI or VSID space to be decoupled between different data center networks without the need for a dedicated Gateway at the edge of the data centers.

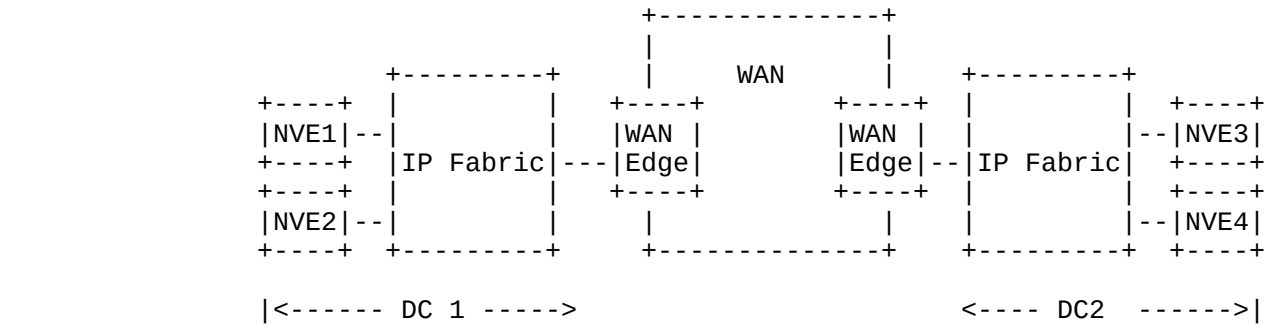


Figure 2: Data Center Interconnect without Gateway

5.1.2 Virtual Identifiers to EVI Mapping

When the EVPN control plane is used in conjunction with VXLAN or NVGRE, two options for mapping the VXLAN VNI or NVGRE VSID to an EVI are possible:

1. Option 1: Single Subnet per EVI

In this option, a single subnet represented by a VNI or VSID is mapped to a unique EVI. This corresponds to the VLAN Based service in [RFC7432], where a tenant VLAN ID gets mapped to an EVPN instance (EVI). As such, a BGP RD and RT is needed per VNI / VSID on every VTEP. The advantage of this model is that it allows the BGP RT constraint mechanisms to be used in order to limit the propagation and import of routes to only the VTEPs that are interested in a given VNI or VSID. The disadvantage of this model may be the provisioning overhead if RD and RT are not derived automatically from VNI or VSID.

In this option, the MAC-VRF table is identified by the RT in the control plane and by the VNI or VSID in the data-plane. In this option, the specific the MAC-VRF table corresponds to only a single bridge table.

2. Option 2: Multiple Subnets per EVI

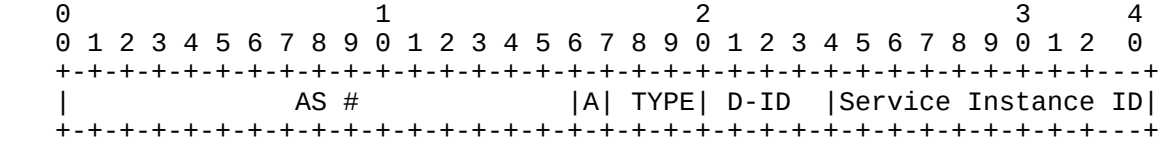
In this option, multiple subnets each represented by a unique VNI or VSID are mapped to a single EVI. For example, if a tenant has multiple segments/subnets each represented by a VNI or VSID, then all the VNIs (or VSIDs) for that tenant are mapped to a single EVI - e.g., the EVI in this case represents the tenant and not a subnet . This corresponds to the VLAN-Aware Bundle service in [RFC7432]. The advantage of this model is that it doesn’t require the provisioning of RD/RT per VNI or VSID. However, this is a moot point if option 1 with auto-derivation is used. The disadvantage of this model is that routes would be imported by VTEPs that may not be interested in a given VNI or VSID.

In this option the MAC-VRF table is identified by the RT in the control plane and a specific bridge table for that MAC-VRF is identified by the <RT, Ethernet Tag ID> in the control plane. In this option, the VNI/VSID in the data-plane is sufficient to identify a specific bridge table - e.g., no need to do a lookup based on VNI/VSID and Ethernet Tag ID fields to identify a bridge table.

5.1.2.1 Auto Derivation of RT

When the option of a single VNI or VSID per EVI is used, it is important to auto-derive RT for EVPN BGP routes in order to simplify configuration for data center operations. RD can be derived easily as described in [RFC7432] and RT can be auto-derived as described next.

Since a gateway PE as depicted in figure-1 participates in both the DCN and WAN BGP sessions, it is important that when RT values are auto-derived for VNIs (or VSIDs), there is no conflict in RT spaces between DCN and WAN networks assuming that both are operating within the same AS. Also, there can be scenarios where both VXLAN and NVGRE encapsulations may be needed within the same DCN and their corresponding VNIs and VSIDs are administered independently which means VNI and VSID spaces can overlap. In order to ensure that no such conflict in RT spaces arises, RT values for DCNs are auto-derived as follow:



- 2 bytes of global admin field of the RT is set to the AS number.

- Three least significant bytes of the local admin field of the RT is set to the VNI or VSID, I-SID, or VID. The most significant bit of the local admin field of the RT is set as follow:
  0: auto-derived
  1: manually-derived

- The next 3 bits of the most significant byte of the local admin field of the RT identifies the space in which the other 3 bytes are defined. The following spaces are defined:
  0 : VID
  1 : VXLAN
  2 : NVGRE
  3 : I-SID
  4 : EVI
  5 : dual-VID

- The remaining 4 bits of the most significant byte of the local admin field of the RT identifies the domain-id. The default value of domain-id is zero indicating that only a single numbering space exist for a given technology. However, if there are more than one number space exist for a given technology (e.g., overlapping VXLAN spaces), then each of the number spaces need to be identify by their corresponding domain-id starting from 1.

5.1.3 Constructing EVPN BGP Routes

In EVPN, an MPLS label is distributed by the egress PE via the EVPN control plane and is placed in the MPLS header of a given packet by the ingress PE. This label is used upon receipt of that packet by the egress PE for disposition of that packet. This is very similar to the use of the VNI or VSID by the egress VTEP or NVE, respectively, with the difference being that an MPLS label has local significance while a VNI or VSID typically has global significance. Accordingly, and specifically to support the option of locally assigned VNIs, the MPLS label field in the MAC Advertisement, Ethernet AD per EVI, and Inclusive Multicast Ethernet Tag routes is used to carry the VNI or VSID. For the balance of this memo, the MPLS label field will be referred to as the VNI/VSID field. The VNI/VSID field is used for both local and global VNIs/VSIDs, and for either case the entire 24-bit field is used to encode the VNI/VSID value.

For the VLAN-based service (a single VNI per MAC-VRF), the Ethernet Tag field in the MAC/IP Advertisement, Ethernet AD per EVI, and Inclusive Multicast route MUST be set to zero just as in the VLAN Based service in [RFC7432].

For the VLAN-aware bundle service (multiple VNIs per MAC-VRF with each VNI associated with its own bridge table), the Ethernet Tag field in the MAC Advertisement, Ethernet AD per EVI, and Inclusive Multicast route MUST identify a bridge table within a MAC-VRF and the set of Ethernet Tags for that EVI needs to be configured consistently on all PEs within that EVI. For local VNIs, the value advertised in the Ethernet Tag field MUST be set to a VID just as in the VLAN-aware bundle service in [RFC7432]. Such setting must be done consistently on all PE devices participating in that EVI within a given domain. For global VNIs, the value advertised in the Ethernet Tag field SHOULD be set to a VNI as long as it matches the existing semantics of the Ethernet Tag, i.e., it identifies a bridge table within a MAC-VRF and the set of VNIs are configured consistently on each PE in that EVI.

In order to indicate that which type of data plane encapsulation (i.e., VXLAN, NVGRE, MPLS, or MPLS in GRE) is to be used, the BGP Encapsulation extended community defined in [RFC5512] is included with all EVPN routes (i.e. MAC Advertisement, Ethernet AD per EVI, Ethernet AD per ESI, Inclusive Multicast Ethernet Tag, and Ethernet Segment) advertised by an egress PE. Five new values have been assigned by IANA to extend the list of encapsulation types defined in [RFC5512]:

+ 8 - VXLAN Encapsulation
+ 9 - NVGRE Encapsulation
+ 10 - MPLS Encapsulation

+ 11 - MPLS in GRE Encapsulation
+ 12 - VXLAN GPE Encapsulation

If the BGP Encapsulation extended community is not present, then the default MPLS encapsulation or a statically configured encapsulation is assumed.

The Ethernet Segment and Ethernet AD per ESI routes MAY be advertised with multiple encapsulation types as long as they use the same EVPN multi-homing procedures - e.g., the mix of VXLAN and NVGRE encapsulation types is a valid one but not the mix of VXLAN and MPLS encapsulation types.

The Next Hop field of the MP_REACH_NLRI attribute of the route MUST be set to the IPv4 or IPv6 address of the NVE. The remaining fields in each route are set as per [RFC7432].

5.2 MPLS over GRE

The EVPN data-plane is modeled as an EVPN MPLS client layer sitting over an MPLS PSN tunnel. Some of the EVPN functions (split-horizon, aliasing, and backup-path) are tied to the MPLS client layer. If MPLS over GRE encapsulation is used, then the EVPN MPLS client layer can be carried over an IP PSN tunnel transparently. Therefore, there is no impact to the EVPN procedures and associated data-plane operation.

The existing standards for MPLS over GRE encapsulation as defined by [RFC4023] can be used for this purpose; however, when it is used in conjunction with EVPN the key field SHOULD be present, and SHOULD be used to provide a 32-bit entropy field. The Checksum and Sequence Number fields are not needed and their corresponding C and S bits MUST be set to zero.

6 EVPN with Multiple Data Plane Encapsulations

The use of the BGP Encapsulation extended community allows each PE in a given EVI to know each of the encapsulations supported by each of the other PEs in that EVI. I.e., each of the PEs in a given EVI may support multiple data plane encapsulations. An ingress PE can send a frame to an egress PE only if the set of encapsulations advertised by the egress PE in the subject MAC/IP Advertisement or per EVI Ethernet AD route, forms a non-empty intersection with the set of encapsulations supported by the ingress PE, and it is at the discretion of the ingress PE which encapsulation to choose from this intersection. (As noted in section 5.1.3, if the BGP Encapsulation

extended community is not present, then the default MPLS encapsulation or a statically configured encapsulation is assumed.)

An ingress node that uses shared multicast trees for sending broadcast or multicast frames MUST maintain distinct trees for each different encapsulation type.

It is the responsibility of the operator of a given EVI to ensure that all of the PEs in that EVI support at least one common encapsulation. If this condition is violated, it could result in service disruption or failure. The use of the BGP Encapsulation extended community provides a method to detect when this condition is violated but the actions to be taken are at the discretion of the operator and are outside the scope of this document.

7 NVE Residing in Hypervisor

When a PE and its CEs are co-located in the same physical device, e.g., when the PE resides in a server and the CEs are its VMs, the links between them are virtual and they typically share fate; i.e., the subject CEs are typically not multi-homed or if they are multi-homed, the multi-homing is a purely local matter to the server hosting the VM, and need not be "visible" to any other PEs, and thus does not require any specific protocol mechanisms. The most common case of this is when the NVE resides in the hypervisor.

In the sub-sections that follow, we will discuss the impact on EVPN procedures for the case when the NVE resides on the hypervisor and the VXLAN or NVGRE encapsulation is used.

7.1 Impact on EVPN BGP Routes & Attributes for VXLAN/NVGRE Encapsulation

In the scenario where all data centers are under a single administrative domain, and there is a single global VNI/VSID space, the RD MAY be set to zero in the EVPN routes. However, in the scenario where different groups of data centers are under different administrative domains, and these data centers are connected via one or more backbone core providers as described in [NOV3-Framework], the RD must be a unique value per EVI or per NVE as described in [RFC7432]. In other words, whenever there is more than one administrative domain for global VNI or VSID, then a non-zero RD MUST be used, or whenever the VNI or VSID value have local significance, then a non-zero RD MUST be used. It is recommend to use a non-zero RD at all time.

When the NVEs reside on the hypervisor, the EVPN BGP routes and attributes associated with multi-homing are no longer required. This reduces the required routes and attributes to the following subset of

four out of the set of eight :

- MAC Advertisement Route
- Inclusive Multicast Ethernet Tag Route
- MAC Mobility Extended Community
- Default Gateway Extended Community

However, as noted in section 8.6 of [RFC7432] in order to enable a single-homing ingress PE to take advantage of fast convergence, aliasing, and backup-path when interacting with multi-homed egress PEs attached to a given Ethernet segment, a single-homing ingress PE SHOULD be able to receive and process Ethernet AD per ES and Ethernet AD per EVI routes."

7.2 Impact on EVPN Procedures for VXLAN/NVGRE Encapsulation

When the NVEs reside on the hypervisors, the EVPN procedures associated with multi-homing are no longer required. This limits the procedures on the NVE to the following subset of the EVPN procedures:

1. Local learning of MAC addresses received from the VMs per section 10.1 of [RFC7432].

2. Advertising locally learned MAC addresses in BGP using the MAC Advertisement routes.

3. Performing remote learning using BGP per Section 10.2 of [RFC7432].

4. Discovering other NVEs and constructing the multicast tunnels using the Inclusive Multicast Ethernet Tag routes.

5. Handling MAC address mobility events per the procedures of Section 16 in [RFC7432].

However, as noted in section 8.6 of [RFC7432] in order to enable a single-homing ingress PE to take advantage of fast convergence, aliasing, and back-up path when interacting with multi-homed egress PEs attached to a given Ethernet segment, a single-homing ingress PE SHOULD implement the ingress node processing of Ethernet AD per ES and Ethernet AD per EVI routes as defined in sections 8.2 Fast Convergence and 8.4 Aliasing and Backup-Path of [RFC7432].

8 NVE Residing in ToR Switch

In this section, we discuss the scenario where the NVEs reside in the Top of Rack (ToR) switches AND the servers (where VMs are residing)

are multi-homed to these ToR switches. The multi-homing may operate in All-Active or Single-Active redundancy mode. If the servers are single-homed to the ToR switches, then the scenario becomes similar to that where the NVE resides in the hypervisor, as discussed in Section 5, as far as the required EVPN functionality.

[RFC7432] defines a set of BGP routes, attributes and procedures to support multi-homing. We first describe these functions and procedures, then discuss which of these are impacted by the encapsulation (such as VXLAN or NVGRE) and what modifications are required.

8.1 EVPN Multi-Homing Features

In this section, we will recap the multi-homing features of EVPN to highlight the encapsulation dependencies. The section only describes the features and functions at a high-level. For more details, the reader is to refer to [RFC7432].

8.1.1 Multi-homed Ethernet Segment Auto-Discovery

EVPN NVEs (or PEs) connected to the same Ethernet Segment (e.g. the same server via LAG) can automatically discover each other with minimal to no configuration through the exchange of BGP routes.

8.1.2 Fast Convergence and Mass Withdraw

EVPN defines a mechanism to efficiently and quickly signal, to remote NVEs, the need to update their forwarding tables upon the occurrence of a failure in connectivity to an Ethernet segment (e.g., a link or a port failure). This is done by having each NVE advertise an Ethernet A-D Route per Ethernet segment for each locally attached segment. Upon a failure in connectivity to the attached segment, the NVE withdraws the corresponding Ethernet A-D route. This triggers all NVEs that receive the withdrawal to update their next-hop adjacencies for all MAC addresses associated with the Ethernet segment in question. If no other NVE had advertised an Ethernet A-D route for the same segment, then the NVE that received the withdrawal simply invalidates the MAC entries for that segment. Otherwise, the NVE updates the next-hop adjacencies to point to the backup NVE(s).

8.1.3 Split-Horizon

If a server is multi-homed to two or more NVEs on an Ethernet segment ES1 operating in all-active redundancy mode sends a multicast, broadcast or unknown unicast packet to a one of these NVEs, then it is important to ensure the packet is not looped back to the server via another NVE connected to this server. The filtering mechanism on

the NVE to prevent such loop and packet duplication is called "split horizon filtering’.

8.1.4 Aliasing and Backup-Path

In the case where a station is multi-homed to multiple NVEs, it is possible that only a single NVE learns a set of the MAC addresses associated with traffic transmitted by the station. This leads to a situation where remote NVEs receive MAC advertisement routes, for these addresses, from a single NVE even though multiple NVEs are connected to the multi-homed station. As a result, the remote NVEs are not able to effectively load-balance traffic among the NVEs connected to the multi-homed Ethernet segment. This could be the case, for e.g. when the NVEs perform data-path learning on the access, and the load-balancing function on the station hashes traffic from a given source MAC address to a single NVE. Another scenario where this occurs is when the NVEs rely on control plane learning on the access (e.g. using ARP), since ARP traffic will be hashed to a single link in the LAG.

To alleviate this issue, EVPN introduces the concept of Aliasing. This refers to the ability of an NVE to signal that it has reachability to a given locally attached Ethernet segment, even when it has learnt no MAC addresses from that segment. The Ethernet A-D route per EVI is used to that end. Remote NVEs which receive MAC advertisement routes with non-zero ESI SHOULD consider the MAC address as reachable via all NVEs that advertise reachability to the relevant Segment using Ethernet A-D routes with the same ESI and with the Single-Active flag reset.

Backup-Path is a closely related function, albeit it applies to the case where the redundancy mode is Single-Active. In this case, the NVE signals that it has reachability to a given locally attached Ethernet Segment using the Ethernet A-D route as well. Remote NVEs which receive the MAC advertisement routes, with non-zero ESI, SHOULD consider the MAC address as reachable via the advertising NVE. Furthermore, the remote NVEs SHOULD install a Backup-Path, for said MAC, to the NVE which had advertised reachability to the relevant Segment using an Ethernet A-D route with the same ESI and with the Single-Active flag set.

8.1.5 DF Election

If a CE is multi-homed to two or more NVEs on an Ethernet segment operating in all-active redundancy mode, then for a given EVI only one of these NVEs, termed the Designated Forwarder (DF) is

responsible for sending it broadcast, multicast, and, if configured for that EVI, unknown unicast frames.

This is required in order to prevent duplicate delivery of multi-destination frames to a multi-homed host or VM, in case of all-active redundancy.

In NVEs where .1Q tagged frames are received from hosts, the DF election is performed on host VLAN IDs (VIDs). It is assumed that for a given Ethernet Segment, VIDs are unique and consistent (e.g., no duplicate VIDs exist).

In GWs where VxLAN encapsulated frames are received, the DF election is performed on VNIs. Again, it is assumed that for a given Ethernet Segment, VNIs are unique and consistent (e.g., no duplicate VNIs exist).

8.2 Impact on EVPN BGP Routes & Attributes

Since multi-homing is supported in this scenario, then the entire set of BGP routes and attributes defined in [RFC7432] are used. As discussed in Section 3.1.3, the VSID or VNI is carried in the VNI/VSID field in the MAC Advertisement, Ethernet AD per EVI, and Inclusive Multicast Ethernet Tag routes.

8.3 Impact on EVPN Procedures

Two cases need to be examined here, depending on whether the NVEs are operating in Active/Standby or in All-Active redundancy.

First, lets consider the case of Active/Standby redundancy, where the hosts are multi-homed to a set of NVEs, however, only a single NVE is active at a given point of time for a given VNI or VSID. In this case, the split-horizon and the aliasing functions are not required but other functions such as multi-homed Ethernet segment auto-discovery, fast convergence and mass withdraw, backup path, and DF election are required.

Second, let’s consider the case of All-Active redundancy. In this case, out of the EVPN multi-homing features listed in section 8.1, the use of the VXLAN or NVGRE encapsulation impacts the split-horizon and aliasing features, since those two rely on the MPLS client layer. Given that this MPLS client layer is absent with these types of encapsulations, alternative procedures and mechanisms are needed to provide the required functions. Those are discussed in detail next.

8.3.1 Split Horizon

In EVPN, an MPLS label is used for split-horizon filtering to support active/active multi-homing where an ingress NVE adds a label corresponding to the site of origin (aka ESI Label) when encapsulating the packet. The egress NVE checks the ESI label when attempting to forward a multi-destination frame out an interface, and if the label corresponds to the same site identifier (ESI) associated with that interface, the packet gets dropped. This prevents the occurrence of forwarding loops.

Since the VXLAN or NVGRE encapsulation does not include this ESI label, other means of performing the split-horizon filtering function MUST be devised. The following approach is recommended for split-horizon filtering when VXLAN or NVGRE encapsulation is used.

Every NVE track the IP address(es) associated with the other NVE(s) with which it has shared multi-homed Ethernet Segments. When the NVE receives a multi-destination frame from the overlay network, it examines the source IP address in the tunnel header (which corresponds to the ingress NVE) and filters out the frame on all local interfaces connected to Ethernet Segments that are shared with the ingress NVE. With this approach, it is required that the ingress NVE performs replication locally to all directly attached Ethernet Segments (regardless of the DF Election state) for all flooded traffic ingress from the access interfaces (i.e. from the hosts). This approach is referred to as "Local Bias", and has the advantage that only a single IP address needs to be used per NVE for split-horizon filtering, as opposed to requiring an IP address per Ethernet Segment per NVE.

In order to prevent unhealthy interactions between the split horizon procedures defined in [RFC7432] and the local bias procedures described in this document, a mix of MPLS over GRE encapsulations on the one hand and VXLAN/NVGRE encapsulations on the other on a given Ethernet Segment is prohibited.

8.3.2 Aliasing and Backup-Path

The Aliasing and the Backup-Path procedures for VXLAN/NVGRE encapsulation is very similar to the ones for MPLS. In case of MPLS, two different Ethernet AD routes are used for this purpose. The one used for Aliasing has a VPN scope and carries a VPN label but the one used for Backup-Path has Ethernet segment scope and doesn’t carry any VPN specific info (e.g., Ethernet Tag and MPLS label are set to zero).

9 Support for Multicast

The E-VPN Inclusive Multicast BGP route is used to discover the multicast tunnels among the endpoints associated with a given VXLAN VNI or NVGRE VSID. The Ethernet Tag field of this route is used to encode the VNI for VLXAN or VSID for NVGRE. The Originating router's IP address field is set to the NVE's IP address. This route is tagged with the PMSI Tunnel attribute, which is used to encode the type of multicast tunnel to be used as well as the multicast tunnel identifier. The tunnel encapsulation is encoded by adding the BGP Encapsulation extended community as per section 3.1.1. The following tunnel types as defined in [RFC6514] can be used in the PMSI tunnel attribute for VXLAN/NVGRE:

+ 3 - PIM-SSM Tree
+ 4 - PIM-SM Tree
+ 5 - BIDIR-PIM Tree
+ 6 - Ingress Replication

Except for Ingress Replication, this multicast tunnel is used by the PE originating the route for sending multicast traffic to other PEs, and is used by PEs that receive this route for receiving the traffic originated by CEs connected to the PE that originated the route.

In the scenario where the multicast tunnel is a tree, both the Inclusive as well as the Aggregate Inclusive variants may be used. In the former case, a multicast tree is dedicated to a VNI or VSID. Whereas, in the latter, a multicast tree is shared among multiple VNIs or VSIDs. This is done by having the NVEs advertise multiple Inclusive Multicast routes with different VNI or VSID encoded in the Ethernet Tag field, but with the same tunnel identifier encoded in the PMSI Tunnel attribute.

10 Data Center Interconnections - DCI

For DCI, the following two main scenarios are considered when connecting data centers running evpn-overlay (as described here) over MPLS/IP core network:

- Scenario 1: DCI using GWs
- Scenario 2: DCI using ASBRs

The following two subsections describe the operations for each of these scenarios.

10.1 DCI using GWs

This is the typical scenario for interconnecting data centers over WAN. In this scenario, EVPN routes are terminated and processed in

each GW and MAC/IP routes are always re-advertised from DC to WAN but from WAN to DC, they are not re-advertised if unknown MAC address (and default IP address) are utilized in NVEs. In this scenario, each GW maintains a MAC-VRF (and/or IP-VRF) for each EVI. The main advantage of this approach is that NVEs do not need to maintain MAC and IP addresses from any remote data centers when default IP route and unknown MAC routes are used - i.e., they only need to maintain routes that are local to their own DC. When default IP route and unknown MAC route are used, any unknown IP and MAC packets from NVEs are forwarded to the GWs where all the VPN MAC and IP routes are maintained. This approach reduces the size of MAC-VRF and IP-VRF significantly at NVEs. Furthermore, it results in a faster convergence time upon a link or NVE failure in a multi-homed network or device redundancy scenario, because the failure related BGP routes (such as mass withdraw message) do not need to get propagated all the way to the remote NVEs in the remote DCs. This approach is described in details in section 3.4 of [DCI-EVPN-OVERLAY].

10.2 DCI using ASBRs

This approach can be considered as the opposite of the first approach and it favors simplification at DCI devices over NVEs such that larger MAC-VRF (and IP-VRF) tables are need to be maintained on NVEs; whereas, DCI devices don’t need to maintain any MAC (and IP) forwarding tables. Furthermore, DCI devices do not need to terminate and processed routes related to multi-homing but rather to relay these messages for the establishment of an end-to-end LSP path. In other words, DCI devices in this approach operate similar to ASBRs for inter-AS options B. This requires locally assigned VNIs to be used just like downstream assigned MPLS VPN label where for all practical purposes the VNIs function like 24-bit VPN labels. This approach is equally applicable to data centers (or access networks) with MPLS encapsulation.

In inter-AS option B, when ASBR receives an EVPN route from its DC over iBGP and re-advertises it to other ASBRs, it re-advertises the EVPN route by re-writing the BGP next-hops to itself, thus losing the identity of the PE that originated the advertisement. This re-write of BGP next-hop impacts the EVPN Mass Withdraw route (Ethernet A-D per ES) and its procedure adversely. In EVPN, the route used for aliasing (Ethernet A-D per EVI route) has the same RD as the MAC/IP routes associated with that EVI. Therefore, the receiving PE can associated the receive MAC/IP routes with its corresponding aliasing route using their RDs even if their next hop is written to the same ASBR router’s address. However, in EVPN, the mass-withdraw route uses a different RD than that of its associated MAC/IP routes. Thus, the way to associate them together is via their next-hop router’s address. Now, when BGP next hop address representing the originating

PE, gets re-written by the re-advertising ASBR, it creates ambiguity in the receiving PE that cannot be resolved. Therefore, the functionality needed at the ASBRs depends on whether the EVPN Ethernet A-D routes (per ES and/or per EVI) are originated and whether there is a need to handle route resolution ambiguity for Ethernet A-D per ES route.

The following two subsections describe the functionality needed by the ASBRs depending on whether the NVEs reside in a Hypervisors or in TORs.

10.2.1 ASBR Functionality with NVEs in Hypervisors

When NVEs reside in hypervisors as described in section 7.1, there is no multi-homing and thus there is no need for the originating NVE to send Ethernet A-D per ES or Ethernet A-D per EVI routes. Furthermore, the processing of these routes by the receiving NVE in the hypervisor are optional per [RFC7432] and as described in section 7. Therefore, the ambiguity issue discussed above doesn’t exist for this scenario and the functionality of ASBRs are that of existing L2VPN (or L3VPN) where the ASBRs assist in setting up end-to-end LSPs among the NVEs’ MAC-VRFs. As noted previously, for all practical purposes, the 24-bit locally assigned VNIs used in this scenario, function as 24-bit labels in setting up the end-to-end LSPs.

10.2.2 ASBR Functionality with NVEs in TORs

When NVEs reside in TORs and operate in multi-homing redundancy mode, then as described in section 8, there is a need for the originating NVE to send Ethernet A-D per ES route(s) (used for mass withdraw) and Ethernet A-D per EVI routes (used for aliasing). As described above, the re-write of BGP next-hop by ASBRs creates ambiguities when Ethernet A-D per ES routes are received by the remote PE in a different ASBR because the receiving PE cannot associated that route with the MAC/IP routes from the same Ethernet Segment advertised by the same originating PE. This ambiguity inhibits the function of mass-withdraw per ES by the receiving PE in a different ASBR.

As an example consider a scenario where CE is multi-homed to PE1 and PE2 where these PEs are connected via ASBR1 and then ASBR2 to the remote PE3. Furthermore, consider that PE1 receives M1 from CE1 but not PE2. Therefore, PE1 advertises Eth A-D per ES1, Eth A-D per EVI1, and M1; whereas, PE2 only advertises Eth A-D per ES1 and Eth A-D per EVI1. ASBR1 receives all these five advertisements and passes them to ASBR2 (with itself as the BGP next hop). ASBR2, in turn, passes them to the remote PE3 with itself as the BGP next hop. PE3 receives these five routes where all of them have the same BGP next-hop (i.e., ASBR2). Furthermore, the two Ether A-D per ES routes received by PE3

have the same info - i.e., same ESI and the same BGP next hop. Although both of these routes are maintained by the BGP process in PE3, information from only one of them is used in the L2 routing table (L2 RIB).

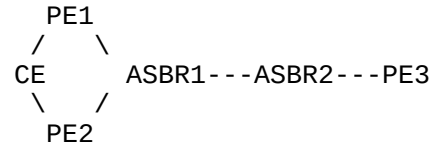


Figure 1: Inter-AS Option B

Now, when the AC between the PE2 and the CE fails and PE2 sends NLRI withdrawal for Ether A-D per ES route and this withdrawal gets propagated and received by the PE3, the BGP process in PE3 removes the corresponding BGP route; however, it doesn't remove the associated info (namely ESI and BGP next hop) from the L2 routing table (L2 RIB) because it still has the other Ether A-D per ES route (originated from PE1) with the same info. That is why the mass-withdraw mechanism does not work when doing DCI with inter-AS option B. However, as described next, the Aliasing function works and so does mass-withdraw per EVI (which is associated with withdrawing the EVPN route associated with Aliasing - i.e., Ether A-D per EVI route).

In the above example, the PE3 receives two Aliasing routes with the same BGP next hop (ASBR2) but different RDs. One of the Alias route has the same RD as the advertised MAC route (M1). PE3 follows the route resolution procedure specified in [RFC7432] upon receiving the two Aliasing route. PE3 should also resolve the alias path properly even though both the primary and backup paths have the same BGP next hop, they have different RDs and the alias route with the different RD than that of the MAC route is considered as the backup path. Therefore, PE3 installs both primary and backup paths (and their associated ESI/EVI MPLS labels or local VNIs) for the MAC route M1. This creates two end-to-end LSPs from PE3 to PE1 for M1 such that when PE3 wants to forward traffic destined to M1, it can load balanced between the two paths. Although route resolution for Aliasing routes with the same BGP next hop is not described in this level of details in [RFC7432], it is expected to operate as such and thus it is clarified here.

When the AC between the PE2 and the CE fails and PE2 sends NLRI withdrawal for Ether A-D per EVI routes and these withdrawals get propagated and received by the PE3, the PE3 removes the Aliasing

route and updates all the corresponding MAC routes for that EVI to remove the backup path. This action makes the mass-withdraw functionality to perform at the per-EVI level (instead of per-ES). The mass-withdraw at per-EVI level requires more messages than that of per-ES level and thus its convergence time is not as good as per ES level. However, its convergence time is much better than individual MAC withdraw.

In summary, it can be seen that aliasing and backup path functionality should work as is for inter-AS option B. Furthermore, in case of inter-AS option B, mass-withdraw functionality falls back from per-ES to per-EVI. If per-ES mass-withdraw functionality is needed along with backward compatibility, then it is recommended to use GWs (per section 10.1) instead of ASBRs for DCI.

11 Acknowledgement

The authors would like to thank David Smith, John Mullooly, Thomas Nadeau for their valuable comments and feedback. The authors would also like to thank Jakob Heitz for his contribution on section 10.

12 Security Considerations

This document uses IP-based tunnel technologies to support data plane transport. Consequently, the security considerations of those tunnel technologies apply. This document defines support for VXLAN and NVGRE encapsulations. The security considerations from those documents as well as [RFC4301] apply to the data plane aspects of this document.

As with [RFC5512], any modification of the information that is used to form encapsulation headers, to choose a tunnel type, or to choose a particular tunnel for a particular payload type may lead to user data packets getting misrouted, misdelivered, and/or dropped.

More broadly, the security considerations for the transport of IP reachability information using BGP are discussed in [RFC4271] and [RFC4272], and are equally applicable for the extensions described in this document.

If the integrity of the BGP session is not itself protected, then an imposter could mount a denial-of-service attack by establishing numerous BGP sessions and forcing an IPsec SA to be created for each one. However, as such an imposter could wreak havoc on the entire routing system, this particular sort of attack is probably not of any special importance.

It should be noted that a BGP session may itself be transported over an IPsec tunnel. Such IPsec tunnels can provide additional security to a BGP session. The management of such IPsec tunnels is outside the scope of this document.

13 IANA Considerations

IANA has allocated the following BGP Tunnel Encapsulation Attribute Tunnel Types:

8	VXLAN Encapsulation
9	NVGRE Encapsulation
10	MPLS Encapsulation
11	MPLS in GRE Encapsulation
12	VXLAN GPE Encapsulation

14 References

14.1 Normative References

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

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

[RFC4272] S. Murphy, "BGP Security Vulnerabilities Analysis.", January 2006.

[RFC4301] S. Kent, K. Seo., "Security Architecture for the Internet Protocol.", December 2005.

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, April 2009.

[RFC7432] Sajassi et al., "BGP MPLS Based Ethernet VPN", RFC 7432, February 2014

14.2 Informative References

[RFC7209] Sajassi et al., "Requirements for Ethernet VPN (EVPN)", RFC 7209, May 2014

[RFC7348] Mahalingam, M., et al, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, August

2014

[NVGRE] Garg, P., et al., "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-07.txt, November 11, 2014

[Problem-Statement] Narten et al., "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-01, September 2012.

[L3VPN-ENDSYSTEMS] Marques et al., "BGP-signaled End-system IP/VPNs", draft-ietf-l3vpn-end-system, work in progress, October 2012.

[NOV3-FRWK] Lasserre et al., "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-01.txt, work in progress, October 2012.

Contributors

S. Salam K. Patel D. Rao S. Thoria D. Cai Cisco

Y. Rekhter R. Shekhar Wen Lin Nischal Sheth Juniper

L. Yong Huawei

Authors' Addresses

Ali Sajassi
Cisco
Email: sajassi@cisco.com

John Drake
Juniper Networks
Email: jdrake@juniper.net

Nabil Bitar
Verizon Communications
Email : nabil.n.bitar@verizon.com

Aldrin Isaac
Juniper
Email: aisaac@juniper.net

James Uttaro
AT&T
Email: uttaro@att.com

Wim Henderickx
Alcatel-Lucent
e-mail: wim.henderickx@alcatel-lucent.com

L2VPNs
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2016

K. Patel
A. Sajassi
Cisco Systems
J. Drake
Juniper Networks, Inc.
W. Henderickx
Alcatel-Lucent

July 2, 2015

Virtual Hub-and-Spoke in BGP EVPNs
draft-keyupate-evpn-virtual-hub-00

Abstract

Ethernet Virtual Private Network (EVPN) solution is becoming pervasive for Network Virtualization Overlay (NVO) services in data center (DC) applications and as the next generation virtual private LAN services in service provider (SP) applications.

The use of host IP default route and host unknown MAC route within a DC is well understood in order to ensure that leaf nodes within a DC only learn and store host MAC and IP addresses for that DC. All other host MAC and IP addresses from remote DCs are learned and stored in DC GW nodes thus alleviating leaf nodes from learning host MAC and IP addresses from the remote DCs.

This draft further optimizes the MAC and IP address learning at the leaf nodes such that a leaf node within a DC only needs to learn and store MAC and IP addresses associated with the sites directly connected to it. A leaf node does not need to learn and store MAC and IP addresses from any other leaf nodes thus reducing the number of learned MACs and IP addresses per EVI substantially.

The modifications provided by this draft updates and extends RFC7024 for BGP EVPN Address Family.

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/.

Copyright Notice

Table of Contents

1. Introduction 3
2. Requirements Language 4
3. Terminology 4
4. Routing Information Exchange for EVPN routes 4
5. EVPN unknown MAC Route 5
5.1. Originating EVPN Unknown MAC Route by a V-Hub 5
5.2. Processing VPN-MAC EVPN unknown Route by a V-SPOKE . . . 5
5.3. Aliasing 5
5.4. Split-Horizon And Mass Withdraw 6
6. Forwarding Considerations 7
6.1. IP-only Forwarding 7
6.2. MAC-only Forwarding - Bridging 7
6.3. MAC and IP Forwarding - IRB 7
7. Handling of the Broadcast and Multicast traffic 8
8. ARP/ND Suppression 8
9. IANA Considerations 9
10. Security Considerations 9
11. Acknowledgements 9
12. Change Log 9
13. References 9
13.1. Normative References 9
13.2. Informative References 10
Authors' Addresses 11

1. Introduction

Ethernet Virtual Private Network (EVPN) solution is becoming pervasive for Network Virtualization Overlay (NVO) services in data center (DC) applications and as the next generation virtual private LAN services in service provider (SP) applications.

With EVPN, providing any-to-any connectivity among sites of a given EVPN Instance (EVI) would require each Provider Edge (PE) router connected to one or more of these sites to hold all the host MAC and IP addresses for that EVI. The use of host IP default route and host unknown MAC route within a DC is well understood in order to alleviate the learning of host MAC and IP addresses to only leaf nodes (PEs) within that DC. All other host MAC and IP addresses from remote DCs are learned and stored in DC GW nodes thus alleviating leaf nodes from learning host MAC and IP addresses from the remote DCs.

This draft further optimizes the MAC and IP address learning at the leaf nodes such that a leaf node within a DC only needs to learn and store MAC and IP addresses associated with the sites directly connected to it. A leaf node does not need to learn and store MAC and IP addresses from any other leaf nodes thus reducing the number of learned MACs and IP addresses per EVI substantially.

[RFC7024] provides rules for Hub and Spoke VPNs for BGP L3VPNs. This draft updates and extends [RFC7024] for BGP EVPN Address Family. This draft provides rules for Originating and Processing of the EVPN host unknown MAC route and host default IP route by EVPN Virtual Hub (V-HUB). This draft also provides rules for the handling of the BUM traffic in Hub and Spoke EVPNs and handling of ARP suppression.

The leaf nodes and DC GW nodes in a data center are referred to as Virtual Spokes (V-spokes) and Virtual Hubs (V-hubs) respectively. A set of V-spoke can be associated with one or more V-hubs. If a V-spokes is associated with more than one V-hubs, then it can load balanced traffic among these V-hubs. Different V-spokes can be associated with different sets of V-hubs such that at one extreme each V-spoke can have a different V-hub set although this may not be desirable and a more typical scenario may be to associate a set of V-spokes to a set of V-hubs - e.g., topology for a DC POD where a set of V-spokes are associated with a set of spine nodes or DC GW nodes.

In order to avoid repeating many of the materials covered in [RFC7024], this draft is written as a delta document with its sections organized to follow those of that RFC with only delta description pertinent to EVPN operation in each section. Therefore,

it is assumed that the readers are very familiar with [RFC7024] and EVPN.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without any normative meaning.

3. Terminology

ARP: Address Resolution Protocol
BEB: Backbone Edge Bridge
B-MAC: Backbone MAC Address
CE: Customer Edge
C-MAC: Customer/Client MAC Address
ES: Ethernet Segment
ESI: Ethernet Segment Identifier
IRB: Integrated Routing and Bridging
LSP: Label Switched Path
MP2MP: Multipoint to Multipoint
MP2P: Multipoint to Point
ND: Neighbor Discovery
NA: Neighbor Advertisement
P2MP: Point to Multipoint
P2P: Point to Point
PE: Provider Edge
EVPN: Ethernet VPN
EVI: EVPN Instance
RT: Route Target

Single-Active Redundancy Mode: When only a single PE, among a group of PEs attached to an Ethernet segment, is allowed to forward traffic to/from that Ethernet Segment, then the Ethernet segment is defined to be operating in Single-Active redundancy mode.

All-Active Redundancy Mode: When all PEs attached to an Ethernet segment are allowed to forward traffic to/from that Ethernet Segment, then the Ethernet segment is defined to be operating in All-Active redundancy mode.

4. Routing Information Exchange for EVPN routes

[RFC7024] defines multiple Route Types NLRI along with procedures for advertisements and processing of these routes. Some of these procedures are impacted as the result of hub-and-spoke architecture.

The routing information exchange among the hub, spoke, and vanilla PEs are subject to the same rules as described in section 3 of [RFC7024]. Furthermore, if there are any changes to the EVPN route advisements and processing from advertisements and processing from [RFC7024], they are described below.

5. EVPN unknown MAC Route

Section 3 of [RFC7024] talks about how a V-hub of a given VPN must export a VPN-IP default route for that VPN and this route must be exported to only the V-spokes of that VPN associated with that V-hub. [I-D.EVPN-overlay] defines the notion of the unknown MAC route for an EVI which is analogous to a VPN-IP default route for a VPN. This unknown MAC route is exported by a V-hub to its associated V-spokes. If multiple V-hubs are associated with a set of V-spokes, then each V- hub advertises it with a distinct RD when originating this route. If a V-spoke imports several of these unknown MAC routes and they all have the same preference, then traffic from the V-spoke to other sites of that EVI would be load balanced among the V-hubs.

5.1. Originating EVPN Unknown MAC Route by a V-Hub

Section 7.3 of the [RFC7024] defines procedures for originating a VPN-IP default route for a VPN. The same procuedures apply when a V-hub wants to originate EVPN unknown MAC route for a given EVI. The V-hub MUST announce unknown MAC route using the MAC/IP advertisement route along with the Default Gateway extended community as defined in section 10.1 of the [RFC7432].

5.2. Processing VPN-MAC EVPN unknown Route by a V-SPOKE

Within a given EVPN, a V-spoke MUST import all the unknown MAC routes unless the route-target mismatch happens. The processing of the received VPN-MAC EVPN default route follows the rules explained in the section 3 of the [RFC7024]. The unknown MAC route MUST be installed according to the rules of MAC/IP Advertisement route installation rules in section 9.2.2 of [RFC7024].

In absense of any more specific VPN-MAC EVPN routes, V-spokes installing the unknown MAC route MUST use the route when performing ARP proxy. This behavior would allow V-Spokes to forward the traffic towards V-Hub.

5.3. Aliasing

[RFC7432] describes the concept and procedures for Aliasing where a station is multi-homed to multiple PEs operating in an All-Active redundancy mode, it is possible that only a single PE learns a set of

MAC addresses associated with traffic transmitted by the station. [RFC7432] describes the concepts and procedures for Aliasing, which occurs when a CE is multi-homed to multiple PE nodes, operating in all-active redundancy mode, but not all of the PEs learn the CE's set of MAC addresses. This leads to a situation where remote PEs receive MAC advertisement routes, for these addresses, from a single NVE even though multiple NVEs are connected to the multi-homed station. As a result, the remote NVEs are not able to effectively load-balance traffic among the NVEs connected to the multi-homed Ethernet segment.

To alleviate this issue, EVPN introduces the concept of Aliasing. This refers to the ability of a PE to signal that it has reachability to a given locally attached Ethernet segment, even when it has learnt no MAC addresses from that segment. The Ethernet A-D per-EVI route is used to that end. Remote PEs which receive MAC advertisement routes with non-zero ESI SHOULD consider the MAC address as reachable via all NVEs that advertise reachability to the relevant Segment using Ethernet A-D routes with the same ESI and with the Single-Active flag reset.

This procedure is impacted for virtual hub-and-spoke topology because a given V-spoke does not receive any MAC/IP advertisements from remote V-spokes; therefore, there is no point in propagating Ethernet A-D per-EVI route to the remote V-spokes. In this solution, the V-hubs terminate the Ethernet A-D per-EVI route (used for Aliasing) and follows the procedures described in [RFC7432] for handling this route.

There are scenarios for which it is desirable to establish direct communication path between a pair of V-spokes for a given host MAC address. In such scenario, the advertising V-spoke advertises both the MAC/IP route and Ethernet A-D per-EVI route with the RT of V-hub (RT-VH) per section 3 of [RFC7024]. The use of RT-VH, ensures that these routes are received by the V-spokes associated with that V-hub set and thus enables the V-spokes to perform the Aliasing procedure.

In summary, PE devices (V-hubs in general and V-spokes occasionally) that receive EVPN MAC/IP route advertisements (associated with a multi-homed site) need to also receive the associated Ethernet A-D per-EVI route advertisement(s) in order for them to perform Aliasing procedure.

5.4. Split-Horizon And Mass Withdraw

[RFC7432] uses Ethernet A-D per-ES route to a) signal to remote PEs the multi-homing redundancy type (Single-Active versus All-Active), b) advertise ESI label for split-horizon filtering when MPLS encapsulation is used, and c) advertise mass-withdraw when a failure

of an access interface impacts many MAC addresses. This route does not need to be advertise from a V-spoke to any remote V-spoke unless a direct communication path between a pair of spoke is needed for a given flow.

Even if communication between a pair of V-spoke is needed for just a single flow, the Ethernet A-D per ES route needs to be advertised from the originating V-spoke for that ES which may handle tens or hundreds of thousands of flows. This is because in order to perform Aliasing function for a given flow, the Ethernet A-D per-EVI route is needed and this route itself is dependent on the Ethernet A-D per-ES route. In such scenario, the advertising V-spoke advertises the Ethernet A-D per-ES route with the RT of V-hub (RT-VH) per section 3 of [RFC7024].

In summary, PE devices (V-hubs in general and V-spokes occasionally) that receive EVPN MAC/IP route advertisements (associated with a multi-homed site) need to also receive the associated Ethernet A-D per-ES route advertisement(s).

6. Forwarding Considerations

6.1. IP-only Forwarding

When EVPN operates in IP-only forwarding mode using EVPN Route Type 5, then all forwarding considerations in section 4 of [RFC7024] are directly applicable here.

6.2. MAC-only Forwarding - Bridging

When EVPN operates in MAC-only forwarding mode (i.e., bridging mode), then for a given EVI, the MPLS label that a V-hub advertises with an Unknown MAC address MUST be the label that identifies the MAC-VRF of the V-hub in absense of a more specific MAC route. When the V-hub receives a packet with such label, the V- hub pops the label and determines further disposition of the packet based on the lookup in the MAC-VRF. Otherwise, the MPLS label of the matching more specific route is used and packet is is forwarded towards the associated NEXTHOP of the more specific route.

6.3. MAC and IP Forwarding - IRB

When a EVPN speaker operates in IRB mode, it implements both the "IP and MAC forwarding Modes" (aka Integrated Routing and Bridging - IRB). On a packet by packet basis, the V-spoke decides whether to do forwarding based on a MAC address lookup (bridge) or based on a IP address lookup (route). If the host destination MAC address is that of the IRB interface (i.e., if the traffic is inter-subnet), then the

V-spoke performs an additional IP lookup in the IP-VRF. However, if the host destination MAC address is that of an actual host MAC address (i.e., the traffic is intra-subnet) , then the V-spoke only performs a MAC lookup in the MAC-VRF. The procedure specified in Section 6.1 and Section 6.2 are applicable to inter-subnet and intra-subnet forwarding respectively. For intra-subnet traffic, if the MAC address is not found in the MAC-VRF, then the V-spoke forwards the traffic to the V-hub with the MPLS label received from the V-hub for the unknown MAC address. For the Inter-subnet traffic, if the IP prefix is not found in the IP-VRF, then the V-spoke forwards the traffic to the V-hub with the MPLS label received from the V-hub for the default IP address.

7. Handling of the Broadcast and Multicast traffic

The handling of the Broadcast and Multicast traffic should be done according to the EVPN rules described in [RFC7432].

8. ARP/ND Suppression

[RFC7432] defines the procedures for ARP/ND suppression where a PE can terminate gratuitous ARP/ND request message from directly connected site and advertises the associated MAC and IP addresses in an EVPN MAC/IP advertisement route to all other remote PEs. The remote PEs that receive this EVPN route advertisement, install the MAC/IP pair in their ARP/ND cache table thus enabling them to terminate ARP/ND requests and generate ARP/ND responses locally thus suppressing the flooding of ARP/ND requests over the EVPN network.

In this hub-and-spoke approach, the ARP suppression needs to be performed by both the EVPN V-hubs as well V-spokes as follow. When a V-Spoke receives a gratuitous ARP/ND request, it terminates it and stores the source MAC/IP pair in its ARP/ND cache table. Then, it advertises the source MAC/IP pair to its associated V-Hubs using EVPN MAC/IP advertisement route. The V-Hubs upon receiving this EVPN route advertisement, create an entry in their ARP/ND cache table for this MAC/IP pair.

Now when a V-Spoke receives an ARP/ND request, it first looks up its ARP cache table, if an entry for that MAC/IP pair is found, then an ARP/ND response is generated locally and sent to the CE. However, if an entry is not found, then the ARP/ND request is unicasted to one of the V-hub associated with this V-spoke. Since, the associated V-hub keeps all the MAC/IP ARP entries in its cache table, it can formulate and ARP/ND response and forward it to that CE via the corresponding V-spoke.

9. IANA Considerations

This document does NOT make any new requests for IANA allocations.

10. Security Considerations

All the security considerations in [RFC7432] apply directly to this document because this document leverages [RFC7432] control plane and their associated procedures - although not the complete set but rather a subset.

This draft does not introduce any new security considerations beyond that of [RFC7432] and [RFC4761] because advertisements and processing of B-MAC addresses follow that of [RFC7432] and processing of C-MAC addresses follow that of [RFC4761] - i.e, B-MAC addresses are learned in control plane and C-MAC addresses are learned in data plane.

11. Acknowledgements

The authors would like to thank Yakov Rekhter for initial idea discussions.

12. Change Log

Initial Version: Sep 21 2014

13. References

13.1. Normative References

[I-D.ietf-l2vpn-evpn] Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-11 (work in progress), October 2014.

[RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

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

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

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

[RFC4374] McCobb, G., "The application/xv+xml Media Type", RFC 4374, January 2006.

[RFC6459] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, January 2012.

[RFC7024] Jeng, H., Uttaro, J., Jalil, L., Decraene, B., Rekhter, Y., and R. Aggarwal, "Virtual Hub-and-Spoke in BGP/MPLS VPNs", RFC 7024, October 2013.

[RFC7432] Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, February 2015.

13.2. Informative References

[I-D.drao-bgp-l3vpn-virtual-network-overlays]
Rao, D., Mullooly, J., and R. Fernando, "Layer-3 virtual network overlays based on BGP Layer-3 VPNs", draft-drao-bgp-l3vpn-virtual-network-overlays-03 (work in progress), July 2014.

[I-D.ietf-bess-evpn-overlay]
Sajassi, A., Drake, J., Bitar, N., Isaac, A., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01 (work in progress), February 2015.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.

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

[RFC7080] Sajassi, A., Salam, S., Bitar, N., and F. Balus, "Virtual Private LAN Service (VPLS) Interoperability with Provider Backbone Bridges", RFC 7080, December 2013.

[RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, May 2014.

Authors' Addresses

Keyur Patel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124 95134
USA

Email: keyupate@cisco.com

Ali Sajassi
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95124 95134
USA

Email: sajassi@cisco.com

John E. Drake
Juniper Networks, Inc.

Email: jdrake@juniper.net

Wim Henderickx
Alcatel-Lucent

Email: wim.henderickx@alcatel-lucent.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2016

Z. Li
S. Zhuang
Huawei Technologies
X. Liu
Ericsson
J. Haas
S. Esale
Juniper Networks
B. Wen
Comcast

October 18, 2015

Yang Data Model for BGP/MPLS IP VPN
draft-li-bess-l3vpn-yang-00

Abstract

This document defines a YANG data model that can be used to configure and manage L3VPN (BGP/MPLS IP VPN).

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 20, 2016.

Table of Contents

1. Introduction 2
2. Definitions and Acronyms 3
3. Design of the L3VPN Model 3
3.1. Overview 3
3.2. VPN Instance Configuration 4
3.2.1. Per-Instance Configuration 4
3.2.2. Address Family Configuration of L3VPN Instance . . . 4
3.3. Yang Tree of L3VPN Yang Model 5
4. L3VPN YANG Model 7
5. IANA Considerations 14
6. Security Considerations 14
7. Normative References 15
Authors' Addresses 15

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 data model that can be used to configure and manage L3VPN (BGP/MPLS IP VPN) [RFC4364].

2. Definitions and Acronyms

AF: Address Family

BGP: Border Gateway Protocol

JSON: JavaScript Object Notation

L3VPN: Layer 3 VPN

NETCONF: Network Configuration Protocol

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

YANG: A data definition language for NETCONF

3. Design of the L3VPN Model

3.1. Overview

The L3VPN Yang module is to augment the routing instance Yang models proposed by the draft [I-D.ietf-netmod-routing-cfg]. It introduced the "l3vpn"container to define augmented parameters which can be applied for VRF Routing Instance and support both the IPv4 and IPv6 address families. The overview of the "l3vpn" container is shown in the following figure:

```
module: ietf-l3vpn
augment /rt:routing/rt:routing-instance:
   +--rw l3vpn
      +--rw ipv4-family
      |  +--rw bgp-parameters
      |  |  +--rw common
      |  |     +--rw route-distinguisher?   string
      |  |     +--rw vpn-targets* [rt-value]
      |  |        +--rw rt-value    string
      |  |        +--rw rt-type     bgp-rt-type
      |  ......
      +--rw ipv6-family
         +--rw bgp-parameters
         |  +--rw common
         |     +--rw route-distinguisher?   string
         |     +--rw vpn-targets* [rt-value]
         |        +--rw rt-value    string
         |        +--rw rt-type     bgp-rt-type
         ......
```

L3VPN inteface parameters can reuse those parameters defined by [I-D.ietf-netmod-routing-cfg].

BGP Protocols parameters for L3VPN is defined by the draft [I-D.ietf-idr-bgp-model]. The augment may be defined in the future version if necessary.

3.2. VPN Instance Configuration

An instance is created to comprise the VPN forwarding information for each VPN in a BGP/MPLS IP VPN. This instance is called a VPN instance or a VPN routing and forwarding (VRF) table. It is also called a per-site forwarding table in [RFC4364]. VPN instances must be created in all BGP/MPLS IP VPN solutions. VPN instances support both the IPv4 and IPv6 address families.

VPN instance configuration consists of the following components :

o Per-Instance Configuration : that contains the common writable configuration objects for VPN instance IPv4 and IPv6 address family.

o Address Family Configuration of L3VPN Instance: that contains the address family specific writable configuration objects.

3.2.1. Per-Instance Configuration

Per-instance parameters is defined by [I-D.ietf-netmod-routing-cfg] including instance name, description, etc.

3.2.2. Address Family Configuration of L3VPN Instance

l3vpn container contains the address family specific writable configuration objects, such as route-distinguisher, vpn-targets, apply-label-mode, etc. The parameters should be consistent between IPv4 family and IPv6 family.

```
      +--rw l3vpn
         +--rw ipv4-family
         |  +--rw bgp-parameters
         |  |  +--rw common
         |  |     +--rw route-distinguisher?   string
         |  |     +--rw vpn-targets* [rt-value]
         |  |        +--rw rt-value    string
         |  |        +--rw rt-type     bgp-rt-type
         |  +--rw apply-label-mode?      apply-label-mode-def
         |  +--rw import-route-policy?   string
         |  +--rw export-route-policy?   string
         |  +--rw tunnel-policy?         string
         |  +--rw prefix-limit
         |  |  +--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
         |  +--rw routing-table-limit
         |  |  +--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
         |  +--rw import-global-rib
         |     +--rw protocol?            enumeration
         |     +--rw processId?           uint32
         |     +--rw bgp-valid-route?     boolean
         |     +--rw route-policy-name?   string
         +--rw ipv6-family
            ......
```

3.3. Yang Tree of L3VPN Yang Model

The Yang tree of L3VPn Yang model is shown in the following figure:

```
module: ietf-l3vpn
augment /rt:routing/rt:routing-instance:
   +--rw l3vpn
      +--rw ipv4-family
      |  +--rw bgp-parameters
      |  |  +--rw common
      |  |     +--rw route-distinguisher?   string
      |  |     +--rw vpn-targets* [rt-value]
      |  |        +--rw rt-value    string
```

```
|  |        +--rw rt-type     bgp-rt-type
|  +--rw apply-label-mode?      apply-label-mode-def
|  +--rw import-route-policy?   string
|  +--rw export-route-policy?   string
|  +--rw tunnel-policy?         string
|  +--rw prefix-limit
|  |  +--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
|  +--rw routing-table-limit
|  |  +--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
|  +--rw import-global-rib
|     +--rw protocol?            enumeration
|     +--rw processId?           uint32
|     +--rw bgp-valid-route?     boolean
|     +--rw route-policy-name?   string
+--rw ipv6-family
   +--rw bgp-parameters
   |  +--rw common
   |     +--rw route-distinguisher?   string
   |     +--rw vpn-targets* [rt-value]
   |        +--rw rt-value    string
   |        +--rw rt-type     bgp-rt-type
   +--rw apply-label-mode?      apply-label-mode-def
   +--rw import-route-policy?   string
   +--rw export-route-policy?   string
   +--rw tunnel-policy?         string
   +--rw prefix-limit
   |  +--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
   +--rw routing-table-limit
   |  +--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
            +--rw import-global-rib
               +--rw protocol?            enumeration
               +--rw processId?           uint32
               +--rw bgp-valid-route?     boolean
               +--rw route-policy-name?   string
```

4. L3VPN YANG Model

```
//L3VPN YANG MODEL
<CODE BEGINS> file "ietf-l3vpn.yang"
module ietf-l3vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l3vpn";
  // replace with IANA namespace when assigned
  prefix "l3vpn";

  import ietf-routing {
    prefix "rt";
    //draft-ietf-netmod-routing-cfg-19
  }

  description
    "This YANG module defines the generic configuration data for
     L3VPN service.

     Terms and Acronyms

     BGP (bgp): Border Gateway Protocol
     IPv4 (ipv4):Internet Protocol Version 4
     IPv6 (ipv6): Internet Protocol Version 6

    ";

  revision 2015-10-09 {
    description
      "Initial revision.";
      reference "RFC4271, RFC4364, RFC4365, RFC4760";
  }

  /* 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";
  }

  typedef apply-label-mode-def {
    type enumeration {
      enum "per-route" {
       value 0;
        description
          "By default, the VPN instance IPv4 address family
           assigns a unique label to each route to be sent
           to the peer PE.";
      }
      enum "per-instance" {
       value 1;
        description
          "The apply-label per-instance command enables the
           one-label-per-VPN-instance mode.";
      }
    }
    description "...";
  }

  grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
      description "Parameters for BGP";
      container common {
        description "Common BGP parameters";
        leaf route-distinguisher {
          type string;
          description "BGP RD";
        }
        list vpn-targets {
          key rt-value;
          description "Route Targets";
          leaf rt-value {
            type string;
            description "Route-Target value";
          }
          leaf rt-type {
            type bgp-rt-type;
```

```
            mandatory true;
            description "Type of RT";
          }
        }
      }
    }
  }



  grouping vpn-af-config {
    description
      "A set of configuration parameters that is applicable to both
       IPv4 and IPv6 address family for a VPN instance .";
    leaf apply-label-mode {
     type apply-label-mode-def;
     default "per-route";
    }

    leaf import-route-policy {
      description
        "The import route-policy command associates a VPN instance enabled
         with the IPv4 or IPv6 address family with an import routing policy.
         Only one import routing policy can be associated with a VPN instance
         enabled with the IPv4 or IPv6 address family. If the import
         route-policy command is run more than once, the latest configuration
         overrides the previous ones.";

        config "true";
        type string {
          length "1..40";
        }
    }

    leaf export-route-policy {
      description
        "The export route-policy command associates a VPN instance enabled
         with the IPv4 or IPv6 address family with an export routing policy.
         Only one export routing policy can be associated with a VPN instance
         enabled with the IPv4 or IPv6 address family. If the export
         route-policy command is run more than once, the latest configuration
         overrides the previous ones.";

      config "true";
      type string {
        length "1..40";
      }
```

```
    }

    leaf tunnel-policy {
      description "tunnel policy name";
      type string;
    }

    container 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 {
        description
          "Specifies the maximum number of prefixes supported in the
           VPN instance IPv4 or IPv6 address family.";

        type uint32 {
          range "1..4294967295";
        }
      }

      choice prefix-limit-action {
        case enable-alert-percent {
          leaf alert-percent-value {
            description
              "Specifies the proportion of the alarm threshold to the
               maximum number of prefixes.";
            type uint8 {
              range "1..100";
            }
          }
          leaf route-unchanged {
            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.";

            config "true";
            type boolean;
```

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

            config "true";
            type boolean;
            default "false";
          }
        }
      }
    }


    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.";

      leaf routing-table-limit-number {
        description
          "Specifies the maximum number of routes supported by a VPN
           instance. ";

        config "true";
        type uint32 {
          range "1..4294967295";
        }
      }
      choice routing-table-limit-action {
        case enable-alert-percent {
          leaf alert-percent-value {
            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.";

             config "true";
             type uint8 {
               range "1..100";
             }
           }
         }
         case enable-simple-alert {
           leaf simple-alert {
             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.";

             config "true";
             type boolean;
           }
         }
       }
     }

     container import-global-rib {
       description
         "Route Leaking from a Global Routing Table into a VRF.";

       leaf protocol {
         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.";

         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:";
          }
        }
      }

      leaf processId {
        description
          "Specifies the process ID if the protocol from routes are
           imported is IS-IS.";

        default "0";
        type uint32 {
          range "0..4294967295";
        }
      }

      leaf bgp-valid-route {
        type boolean;
      }
```

```
      leaf route-policy-name {
        description
          "Policy Id for import routes";
        type string {
        }
      }

    }
  }

  augment "/rt:routing/rt:routing-instance" {
    container l3vpn {
      when "/rt:routing/rt:routing-instance/rt:type = 'vrf-routing-instance'";

        container ipv4-family {
          description
            "The IPv4 address family is enabled for the VPN instance.";

          uses bgp-parameters-grp;
          uses vpn-af-config;
        }

        container ipv6-family {
          description
            "The IPv6 address family is enabled for the VPN instance.";

          uses bgp-parameters-grp;
          uses vpn-af-config;
        }

    } //End of case type

  } //End of augment "/rt:routing/rt:routing-instance"

}
</CODE ENDS>
```

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

This document does not introduce any new security risk.

7. Normative References

[I-D.ietf-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-ietf-idr-bgp-model-00 (work in progress), July 2015.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-20 (work in progress), October 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>.

[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>.

Authors' Addresses

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: xufeng.liu@ericsson.com

Jeffrey Haas
Juniper Networks

Email: jhaas@juniper.net

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

Email: sesale@juniper.net

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2016

Z. Li
S. Zhuang
Huawei Technologies

October 18, 2015

Role-Based State Advertisement for Multicast in MPLS/BGP IP VPNs
draft-li-bess-mvpn-role-state-ad-00

Abstract

The document defines and uses a new BGP attribute called the "Role Discovery attribute" to advertise the role and corresponding primary/backup state for Multicast in MPLS/BGP IP VPNs [RFC6514]. The role-based state advertisement can help optimization of process in MPLS/BGP Multicast VPN.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 20, 2016.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Applications and Requirements 3
3.1. Easing Provision of mLDP P2MP LSP 3
3.2. Reducing Unnecessary Traffic Replication 3
3.3. Local Protection of Egress Nodes 4
3.4. Centralized Multicast Traffic Optimization 4
4. Role Discovery Attribute 5
5. Operations 6
5.1. Advertisement of Role-based State for Root Nodes 6
5.2. Advertisement of Role-based State for Leaf Nodes 6
6. IANA Considerations 7
7. Security Considerations 7
8. Acknowledgements 7
9. References 7
9.1. Normative References 8
9.2. Informative References 8
Authors' Addresses 8

1. Introduction

[RFC6513] defines the protocols and procedures for multicast in the BGP/MPLS IP VPN (Virtual Private Network) [RFC4364] and [RFC6514] describes the BGP encodings and procedures for exchanging the information elements required by Multicast in MPLS/BGP IP VPNs.

In MPLS/BGP Multicast VPN, there is close relation between the multicast service and the tunnel which bears the multicast service. The tunnel can be triggered to setup automatically after the auto-discovery of the leaf PEs. This can facilitate the provision of the tunnels for multicast. Or else, it will take much effort for the provision work which is troublesome and error-prone. Based on the thinking, it is desirable that more information can be advertised along with the auto-discovery route which can optimize the provision of MPLS/BGP Multicast VPN.

This document identifies the applications and requirements to advertise the role and corresponding state for Multicast in MPLS/BGP IP VPNs and defines a new BGP attribute called the "Role Discovery

attribute" to achieve the object. The role-based state advertisement can help optimization of multicast process.

2. Terminology

CE: Customer Edge

PE: Provider Edge

MVPN: Multicast VPN

3. Applications and Requirements

3.1. Easing Provision of mLDP P2MP LSP

The multipoint extensions for Label Distribution Protocol (mLDP) P2MP LSP can be used to bear multicast service in MPLS/BGP MVPN [RFC6514]. It needs to send label mappings to the root to set up the P2MP LSP. So it has to specify the root node address on all leaf nodes for a specific P2MP LSP. In MPLS/BGP MVPN, if the root/leaf role information of the PE in the MVPN can be advertised in the auto-discovery route, the leaf PE can directly trigger mLDP to send label mapping to the ingress PE of the MVPN without explicitly specifying the root address. It can facilitate the provision of the MVPN when mLDP P2MP LSP is used.

3.2. Reducing Unnecessary Traffic Replication

There exist multi-homing scenarios in MPLS/BGP MVPN. As shown in the figure 1, CE2 multi-homes to two PEs (PE2 and PE3). We assume PE1 is the ingress PE and PE2/PE3 are the egress PEs for a specific MPLS/ BGP MVPN. If C-join is always sent from the CE2 to the PE2 for the MVPN, the multicast traffic sent from the PE1 to PE3 will be always dropped since there is not (C-S, C-G) entry in the MVPN on PE3. If PE1 can learn that the remote PE would not forward the multicast traffic to any CE, the bandwidth can be saved in the network for PE1 can stop to setup the ingress replication tunnel or P2MP LSPs to the remote PE or stop to replicate the unnecessary traffic to the tunnels to the remote PE. In order to achieve the object, the primary or backup state for the leaf PE can be advertised. As to a PE in a specific MVPN, the primary state means that it needs to forward the multicast traffic to the CE. The backup state means it would not forward the multicast traffic to any CE.

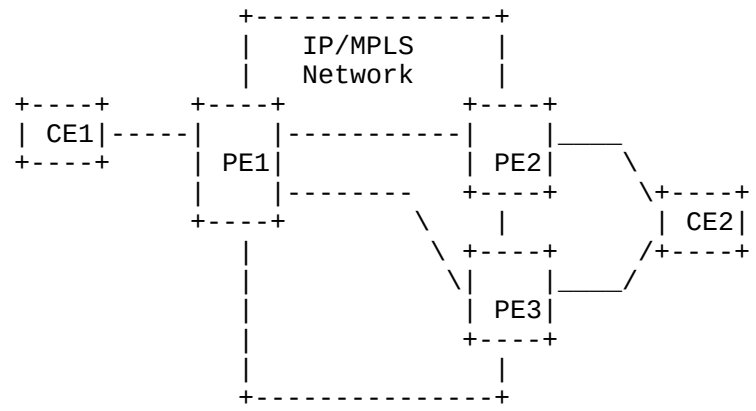


Figure 1 Multi-homing Network for Multicast in MPLS/BGP IP VPN

3.3. Local Protection of Egress Nodes

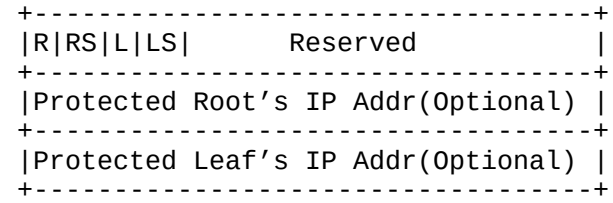
[I-D.ietf-mpls-rsvp-ingress-protection] and [I-D.ietf-mpls-rsvp-egress-protection] proposes mechanisms for locally protecting ingress and egress nodes of MPLS TE P2MP LSPs. In the mechanism for the local protection of egress nodes, the backup egress node needs to be designated for the primary egress node for a P2MP LSP. The previous hop node of the primary egress node sets up a backup Sub-LSP from itself to the backup egress node after receiving the information about the backup egress node. The provision of the local protection mechanism of egress nodes in P2MP LSPs can be facilitated in MPLS/BGP MVPN by advertising the primary/backup state and the protected egress node address with the auto-discovery route. When the ingress PE of the MVPN learns which egress PE can be used as the backup node to protect the primary egress node, it can directly trigger to set up the P2MP LSP with local protection of egress nodes. The method saves much provision effort since it need not statically designate the protection between the backup egress node and the primary egress node for a P2MP LSP.

3.4. Centralized Multicast Traffic Optimization

As the development of central controlled multicast application such as PCE-initiated P2MP LSP [I-D.palle-pce-stateful-pce-initiated-p2mp-lsp], PCE can be used to initiate the setup of RSVP-TE P2MP LSP for the purpose of traffic optimization. In order to support such applications, the controller should learn the roles of Multicast VPN instances distributed on different PEs.

4. Role Discovery Attribute

This document defines and uses a new BGP attribute called the "Role Discovery attribute". This is an optional transitive BGP attribute. The format of this attribute is defined as follows:



R field is one bit to identify if the PE is used as the root node in the MVPN.

RS field is two bits to identify the primary/backup state if the PE is used as the root node. There are three values for the RS field:

- 0 means the PE is used as the primary root node.
- 1 means the PE is used as the backup root node. But the protected root node's IP address does not exist.
- 2 means the PE is used as the backup root node and there exists the protected root node's IP address.

L field is one bit to identify if the PE is used as the leaf node in the MVPN.

LS field is two bits to identify the primary/backup state if the PE is used as the leaf node. There are three values for the LS field:

- 0 means the PE is used as the primary leaf node.
- 1 means the PE is used as the backup leaf node. But the protected leaf node's IP address does not exist.
- 2 means the PE is used as the backup leaf node and there exists the protected leaf node's IP address.

Protected Root's IP Addr is an optional field. It specifies the IPv4/IPv6 address of the protected root node. The field exists only when the value of the RS field is 2.

Protected Leaf's IP Addr is an optional field. It specifies the IPv4/IPv6 address of the protected leaf node. The field exists only when the value of the LS field is 2.

The Role Discovery attribute can be used in conjunction with Intra-AS I-PMSI A-D routes, Inter-AS I-PMSI A-D routes, S-PMSI A-D routes.

5. Operations

5.1. Advertisement of Role-based State for Root Nodes

The Role Discovery attribute can be used in conjunction with Intra-AS I-PMSI A-D routes to advertise the role and corresponding primary/ backup state for root nodes in MPLS/BGP MVPN.

If a PE is specified as the root PE for a specific MVPN, it MUST set the R bit as 1 in the A-D route. Otherwise, the R bit MUST NOT be set.

If the root PE is used as the backup ingress node to protect the primary root PE, the RS field in the A-D route MUST set as 1 when there is no determined root node to be protected. In this case the primary root node protected by the backup root node can be calculated by all nodes according to some uniform algorithms which is out of the scope of this document. When the RS field in the A-D route is set as 1, the Protected Root's IP Addr field MUST NOT exist in the A-D route.

If the root PE is used as the backup ingress node to protect the primary root PE, the RS field in the A-D route MUST set as 2 when there is a determined root node to be protected. In this case the Protected Root's IP Addr field MUST exist in the A-D route which will specify the IPv4/IPv6 address of the protected root node.

If the R bit is set as 1 and the RS field is set as 0 in the A-D route, the Protected Root's IP Addr field MUST NOT exist in the A-D route. This means the PE is used as the primary root PE.

If the R bit in the A-D route is set as 0, the RS field MUST be ignored and the Protected Root's IP Addr field MUST NOT exist in the A-D route.

5.2. Advertisement of Role-based State for Leaf Nodes

The Role Discovery attribute can be used in conjunction with Intra-AS I-PMSI A-D routes to advertise the role and corresponding primary/ backup state for leaf nodes in MPLS/BGP MVPN.

If a PE is specified as the leaf PE for a specific MVPN, it MUST set the L bit as 1 in the A-D route. Otherwise, the L bit MUST NOT be set.

If the leaf PE is used as the backup egress node to protect the primary leaf PE, the LS field in the A-D route MUST set as 1 when there is no determined leaf node to be protected. In this case the primary leaf node protected by the backup leaf node can be calculated by all nodes according to some uniform algorithms which is out of the scope of this document. When the LS field in the A-D route is set as 1, the Protected Leaf's IP Addr field MUST NOT exist in the A-D route.

If the leaf PE is used as the backup egress node to protect the primary leaf PE, the LS field in the A-D route MUST set as 2 when there is a determined leaf node to be protected. In this case the Protected Leaf's IP Addr field MUST exist in the A-D route which will specify the IPv4/IPv6 address of the protected leaf node.

If the L bit is set as 1 and the LS field is set as 0 in the A-D route, the Protected Leaf's IP Addr field MUST NOT exist in the A-D route. This means the PE is used as the primary leaf PE.

If the L bit in the A-D route is set as 0, the LS field MUST be ignored and the Protected Leaf's IP Addr field MUST NOT exist in the A-D route.

6. IANA Considerations

This document defines and uses a new BGP attribute called the "Role Discovery attribute". This is an optional transitive BGP attribute. The type is to be assigned by IANA.

7. Security Considerations

There are no additional security aspects beyond those specified in ([RFC6513]) and [RFC6514].

8. Acknowledgements

The authors would like to thank Hui Ni and Yisong Liu for their contributions to this work

9. References

9.1. Normative References

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

[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>.

[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>.

9.2. Informative References

[I-D.ietf-mpls-rsvp-egress-protection] Chen, H., Li, Z., So, N., Liu, A., Saad, T., Xu, F., Toy, M., Huang, L., and L. Liu, "Extensions to RSVP-TE for LSP Egress Local Protection", draft-ietf-mpls-rsvp-egress-protection-02 (work in progress), October 2014.

[I-D.ietf-mpls-rsvp-ingress-protection] Chen, H. and R. Torvi, "Extensions to RSVP-TE for LSP Ingress Local Protection", draft-ietf-mpls-rsvp-ingress-protection-02 (work in progress), October 2014.

[I-D.palle-pce-stateful-pce-initiated-p2mp-lsp] Palle, U., Dhody, D., Tanaka, Y., Ali, Z., and V. Beeram, "PCEP Extensions for PCE-initiated Point-to-Multipoint LSP Setup in a Stateful PCE Model", draft-palle-pce-stateful-pce-initiated-p2mp-lsp-06 (work in progress), June 2015.

Authors' Addresses

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

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 12, 2016

S. Mohanty
K. Patel
A. Sajassi
Cisco Systems, Inc.
J. Drake
Juniper Networks, Inc.
A. Przygienda
Ericsson

September 9, 2015

A new Designated Forwarder Election for the EVPN
draft-mohanty-bess-evpn-df-election-01

Abstract

This document describes an improved EVPN Designated Forwarder Election (DF) algorithm which can be used to enhance operational experience in terms of convergence speed and robustness over a WAN deploying EVPN

Status of This Memo

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

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

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

This Internet-Draft will expire on March 12, 2016.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Finite State Machine 4
1.2. Requirements Language 4
2. The modulus based DF Election Algorithm 4
3. Problems with the modulus based DF Election Algorithm 5
4. Highest Random Weight 6
5. HRW and Consistent Hashing 7
6. HRW Algorithm for EVPN DF Election 7
7. Protocol Considerations 8
7.1. Finite State Machine 9
8. Operational Considerations 12
9. Security Considerations 12
10. Acknowledgements 12
11. References 12
11.1. Normative References 12
11.2. Informative References 13
Authors' Addresses 14

1. Introduction

Ethernet MPLS VPN (EVPN) [RFC7432] is an emerging technology that is gaining prominence in Internet Service Provider IP/MPLS networks. In EVPN, mac addresses are disseminated as routes across the geographical area via the Border Gateway Protocol, BGP [RFC4271] using the familiar L3VPN model [RFC4364]. An EVPN instance that spans across PEs is defined as an EVI. Constrained Route Distribution [RFC4684] can be used in conjunction to selectively advertise the routes to where they are needed. One of the major advantages of EVPN over VPLS [RFC4761],[RFC6624] is that it provides a solution for minimizing flooding of unknown traffic and also provides all Active mode of operation so that the traffic can truly be multi-homed. In technologies such as EVPN or VPLS, managing Broadcast, Unknown Unicast and multicast traffic (BUM) is a key requirement. In the case where the customer edge (CE) router is multi-homed to one or more Provider Edge (PE) Routers, it is necessary that one and only one of the PE routers should forward BUM traffic into the core or towards the CE as and when appropriate.

Specifically, quoting Section 8.5, [RFC7432], Consider a CE that is a host or a router that is multi-homed directly to more than one PE in an EVPN instance on a given Ethernet segment. One or more Ethernet

Tags may be configured on the Ethernet segment. In this scenario only one of the PEs, referred to as the Designated Forwarder (DF), is responsible for certain actions:

a. Sending multicast and broadcast traffic, on a given Ethernet Tag on a particular Ethernet segment, to the CE.

b. Flooding unknown unicast traffic (i.e. traffic for which an PE does not know the destination MAC address), on a given Ethernet Tag on a particular Ethernet segment to the CE, if the environment requires flooding of unknown unicast traffic.

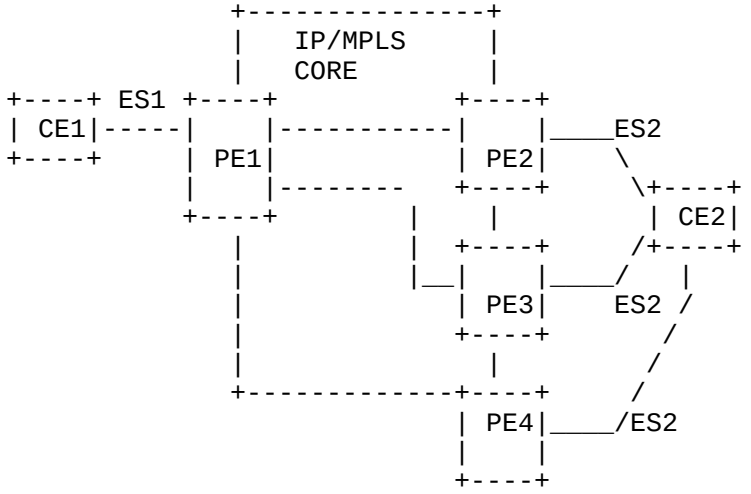


Figure 1 Multi-homing Network of E-VPN

Figure 1

Figure 1 illustrates a case where there are two Ethernet Segments, ES1 and ES2. PE1 is attached to CE1 via Ethernet Segment ES1 whereas PE2, PE3 and PE4 are attached to CE2 via ES2 i.e. PE2, PE3 and PE4 form a redundancy group. Since CE2 is multi-homed to different PEs on the same Ethernet Segment, it is necessary for PE2, PE3 and PE4 to agree on a DF to satisfy the above mentioned requirements.

Layer2 devices are particularly susceptible to forwarding loops because of the broadcast nature of the Ethernet traffic. Therefore it is very important that in case of multi-homing, only one of the links be used to direct traffic to/from the core.

One of the pre-requisites for this support is that participating PEs must agree amongst themselves as to who would act as the Designated Forwarder. This needs to be achieved through a distributed algorithm in which each participating PE independently and unambiguously selects one of the participating PEs as the DF, and the result should be unanimously in agreement.

The DF election algorithm as described in [RFC7432] has some undesirable properties and in some cases can be somewhat disruptive and unfair. This document describes those issues and proposes a mechanism for dealing with those issues. These mechanisms do involve changes to the DF Election algorithm , but do not require any protocol changes to the EVPN Route exchange and have minimal changes to their content per se.

1.1. Finite State Machine

Since the specification in EVPN RFC [RFC7432] does leave several questions open as to the precise final state machine behavior of the DF election, the document also includes a section describing precisely the intended behavior. The finite state machine is presented in Section 7.1

1.2. Requirements Language

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

2. The modulus based DF Election Algorithm

The default procedure for DF election at the granularity of (ESI,EVI) is referred to as "service carving". With service carving, it is possible to elect multiple DFs per Ethernet Segment (one per EVI) in order to perform load-balancing of multi-destination traffic destined to a given Segment. The objective is that the load-balancing procedures should carve up the EVI space among the redundant PE nodes evenly, in such a way that every PE is the DF for a disjoint set of EVIs.

The existing DF algorithm as described in the EVPN RFC(Section 8.5 [RFC7432]) is based on a modulus operation. The PEs to which the ES (for which DF election is to be carried out per vlan) is multi-homed form an ordered (ordinal) list in ascending order of the PE ip address values. Say, there are N PEs, P0, P1, ... PN-1 ranked as per increasing IP addresses in the ordinal list; then for each vlan with ethernet tag v, configured on the ethernet segment ES1, PEx is the DF for vlan v on ES ES1 when x equals (v mod N). In the case when the

vlan density is high meaning there are significant number of vlans and the vlan-id or ethernet-tag is uniformly distributed, the thinking is that the DF election will be spread across the PEs hosting that ethernet segment and good service carving can be achieved.

3. Problems with the modulus based DF Election Algorithm

There are three fundamental problems with the current DF Election.

First, the algorithm will not perform well when the ethernet tag follows a non-uniform distribution, for instance when the ethernet tags are all even or all odd. In such a case let us assume that the ES is multi-homed to two PEs; all the vlans will only pick one of the PEs as the DF. This is very sub-optimal. It defeats the purpose of service carving as the DFs are not really evenly spread across. In this particular case, in fact one of the PEs does not get elected all as the DF, so it does not participate in the DF responsibilities at all. Consider another example where referring to Figure 1, lets assume that PE2, PE3, PE4 are in ascending order of the IP address; and each vlan configured on ES2 is associated with an Ethernet Tag of of the form (3x+1), where x is an integer. This will result in PE3 always be selected as the DF.

Even in the case when the ethernet tag distribution is uniform the instance of a PE being up or down results in re-computation ((v mod N-1) or (v mod N+1) as is the case); The resulting modulus value need not be uniformly distributed but subject to the primality of N-1 or N+1 as may be the case.

The third problem is one of disruption. Consider a case when the same Ethernet Segment is multi homed to a set of PEs. When the ES is down in one of the PEs, say PE1, or PE1 itself reboots, or the BGP process goes down or the connectivity between PE1 and an RR goes down, the effective number of PEs in the system now becomes N-1 and DFs are computed for all the vlans that are configured on that ethernet segment. In general, if the DF for a vlan v happens not to be PE1, but some other PE, say PE2, it is likely that some other PE will become the new DF. This is not desirable. Similarly when a new PE hosts the same Ethernet segment, the mapping again changes because of the mod operation. This results in needless churn. Again referring to Figure 1, say v1, v2 and v3 are vlans configured on ES2 with associated ethernet tags of value 999, 1000 and 10001 respectively. So PE1, PE2 and PE3 are also the DFs for v1, v2 and v3 respectively. Now when PE3 goes down, PE2 will become the DF for v1 and PE1 will become the DF for v2.

One point to note is that the current DF election algorithm assumes that all the PEs who are multi-homed to the same Ethernet Segment and interested in the DF Election by exchanging EVPN routes have a V4 peering with each other or via a Route Reflector. This need not be the case as there can be a v6 peering and supporting the EVPN address-family.

Mathematically, a conventional hash function maps a key k to a number i representing one of m hash buckets through a function h(k) i.e. i=h(k). In the EVPN case, h is simply a modulo-m hash function viz. h(v) = v mod N, where N is the number of PEs that are multi-homed to the Ethernet Segment in discussion. It is well-known that for good hash distribution using the modulus operation, the modulus N should be a prime-number not too close to a power of 2 [CLRS2009]. When the effective number of PEs changes from N to N-1 (or vice versa); all the objects (vlan v) will be remapped except those for which v mod N and v mod (N-1) refer to the same PE in the previous and subsequent ordinal rankings respectively.

From a forwarding perspective, this is a churn, as it results in programming the CE and PE side ports as blocking or non-blocking at potentially all PEs when the DF changes either because (i) a new PE is added or (ii) another one goes down or loses connectivity or else cannot take part in the DF election process for whatever reason. This draft addresses this problem and furnishes a solution to this undesirable behavior.

4. Highest Random Weight

Highest Random Weight (HRW) as defined in [HRW1999] is originally proposed in the context of Internet Caching and proxy Server load balancing. Given an object name and a set of servers, HRW maps a request to a server using the object-name (object-id) and server-name (server-id) rather than the state of the server states. HRW forms a hash out of the server-id and the object-id and forms an ordered list of the servers for the particular object-id. The server for which the hash value is highest, serves as the primary responsible for that particular object, and the server with the next highest value in that hash serves as the backup server. HRW always maps a given object object name to the same server within a given cluster; consequently it can be used at client sites to achieve global consensus on object-server mappings. When that server goes down, the backup server becomes the responsible designate.

Choosing an appropriate hash function that is statistically oblivious to the key distribution and imparts a good uniform distribution of the hash output is an important aspect of the algorithm,. Fortunately many such hash functions exist. [HRW1999] provides pseudorandom

functions based on Unix utilities rand and srand and easily constructed XOR functions that perform considerably well. This imparts very good properties in the load balancing context. Also each server independently and unambiguously arrives at the primary server selection. HRW already finds use in multicast and ECMP [RFC2991],[RFC2992].

In the existing DF algorithm Section 2, whenever a new PE comes up or an existing PE goes down, there is a significant interval before the change is noticed by all peer PEs as it has to be conveyed by the BGP update message involving the type-4 route. There is a timer to batch all the messages before triggering the service carving procedures. When the timer expires, each PE will build the ordered list and follow the procedures for DF Election. In the proposed method which we will describe shortly this "jittered" behavior is retained.

5. HRW and Consistent Hashing

HRW is not the only algorithm that addresses the object to server mapping problem with goals of fair load distribution, redundancy and fast access. There is another family of algorithms that also addresses this problem; these fall under the umbrella of the Consistent Hashing Algorithms [CHASH]. These will not be considered here.

6. HRW Algorithm for EVPN DF Election

The applicability of HRW to DF Election can be described here. Let DF(v) denote the Designated Forwarder and BDF(v) the Backup Designated forwarder for the ethernet tag V, where v is the vlan, Si is the IP address of server i and weight is a pseudorandom function of v and Si. In case of a vlan bundle service, v denotes the lowest vlan similar to the ’lowest vlan in bundle’ logic of [RFC7432].

1. DF(v) = Si: Weight(v, Si) >= Weight(V, Sj) , for all j. In case of a tie, choose the PE whose IP address is numerically the least.

2. BDF(v) = Sk: Weight(v, Si) >= Weight(V, Sk) and Weight(v, Sk) >= Weight(v, Sj). in case of tie choose the PE whose IP address is numerically the least.

Since the Weight is a Pseudorandom function with domain as a concatenation of (v, S), it is an efficient deterministic algorithm which is independent of the Ethernet Tag V sample space distribution. Choosing a good hash function for the pseudorandom function is an important consideration for this algorithm to perform provably better than the existing algorithm. As mentioned previously, such functions

are described in the HRW paper. We take as candidate hash functions two of the ones that are preferred in [HRW1999].

1. Wrand(v, Si) = (1103515245((1103515245.Si+12345)XOR D(v))+12345)(mod 2^31) and

2. Wrand2(v, Si) = (1103515245((1103515245.D(v)+12345)XOR Si)+12345)(mod 2^31)

Here D(v) is the 31-bit digest of the ethernet-tag v and Si is address of the ith server. The server's IP address length does not matter as only the low-order 31 bits are modulo significant. Eventually we plan to choose one of the two candidate hash functions as the preferred one.

A point to note is that the the domain of the Weight function is a concatenation of the ethernet-tag and the PE IP-address, and the actual length of the server IP address (whether V4 or V6) is not really relevant, so long as the actual hash algorithm takes into consideration the concatenated string. The existing algorithm in [RFC7432] as is cannot employ both V4 and V6 neighbor peering address.

HRW solves the disadvantage pointed out in Section 3 and ensures (i) with very high probability that the task of DF election for respective vlans is more or less equally distributed among the PEs even for the 2 PE case (ii)If a PE, hosting some vlans on given ES, but is neither the DF nor the BDF for that vlan, goes down or its connection to the ES goes down, it does not result in a DF and BDF reassignment the other PEs. This saves computation, especially in the case when the connection flaps. (iii)More importantly it avoids the needless disruption case (c) that are inherent in the existing modulus based algorithm (iv)In addition to the DF, the algorithm also furnishes the BDF, which would be the DF if the current DF fails.

7. Protocol Considerations

Note that for the DF election procedures to be globally convergent and unanimous, it is necessary that all the participating PEs agree on the DF Election algorithm to be used. It is not possible that some PEs continue to use the existing modulus based DF election and some newer PEs use the HRW. For brownfield deployments and for interoperability with legacy boxes, its is important that all PEs need to have the capability to fall back on the modulus algorithm. A PE (one with a newer version of the software) can indicate its willingness to support HRW by signaling a new extended community along with the Ethernet-Segment Route (Type-4). This extended community is explained in the next paragraph. When a PE receives the

Ethernet-Segment Routes from all the other PEs for the ethernet segment in question, it checks to see if all the advertisements have the extended community attached; in the case that they do, this particular PE, and by induction all the other PEs proceed to do DF Election as per the HRW Algorithm. Otherwise if even a single advertisement for the type-4 route is not received with the extended community or the received DF types (including locally configured type) do not ALL match a single value, the default modulus algorithm is used as before. Also, the HRW algorithm needs to be executed after the "jittered" time.

A new BGP extended community attribute [RFC4360] needs to be defined to identify the DF election procedure to be used for the Ethernet Segment. We propose to name this extended community as the DF Election Extended Community. It is a new transitive extended community where the Type field is 0x06, and the Sub-Type is to be defined. It may be advertised along with Ethernet Segment routes.

Each DF Election Extended Community is encoded as a 8-octet value as follows:

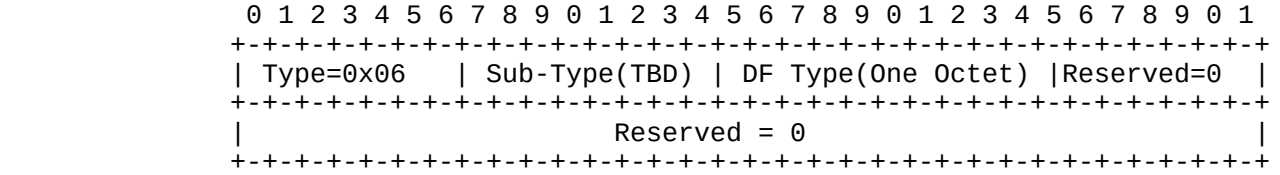


Figure 2

The DF Type state is encoded as one octet. A value of 0 means that the default (the mod based) DF election procedures are used and a value of 1 means that the HRW algorithm will be employed. A request needs to registered with the IETF authority for the subtype [I-D.ietf-idr-extcomm-iana]

7.1. Finite State Machine

Per [RFC7432], the FSM described in Figure 3 is executed per ESI/VLAN in case of VLAN aware service or ESI/[VLANs in VLAN Bundle] in case of VLAN Bundle on each participating PE.

Observe that currently the VLANs are derived from local configuration and the FSM does not provide any protection against misconfiguration where same EVI,ESI combination has different set of VLANs on different participating PEs or one of the PEs elects to consider

VLANs as VLAN bundle and another as separate VLANs for election purposes (service type mismatch).

The FSM is normative in the sense that any design or implementation MUST behave towards external peers and as observable external behavior (DF) in a manner equivalent to this FSM.

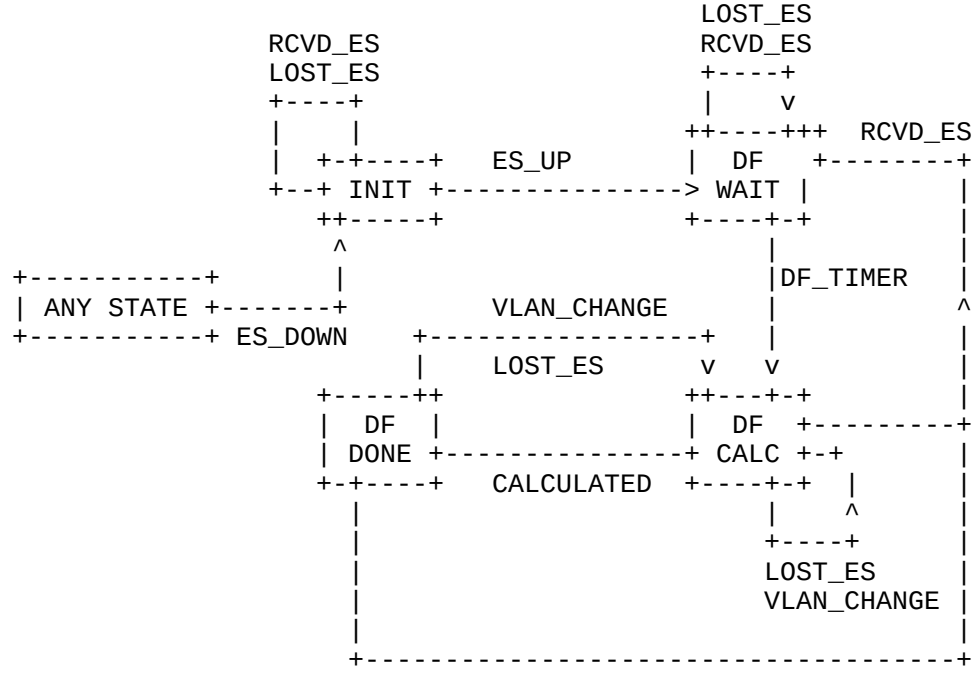


Figure 3

States:

1. INIT: Initial State

2. DF WAIT: State in which the participants waits for enough information to perform the DF election for the EVI/ESI/VLAN combination.

3. DF CALC: State in which the new DF is recomputed.

4. DF DONE: State in which the according DF for the EVI/ESI/VLAN combination has been elected.

Events:

1. ES_UP: The ESI has been locally configured as 'up'.

2. ES_DOWN: The ESI has been locally configured as 'down'.

3. VLAN_CHANGE: The VLANs configured in a bundle that uses the ESI changed. This event is necessary for VLAN bundles only.

4. DF_TIMER: DF Wait timer has expired.

5. RCVD_ES: A new or changed Ethernet Segment Route is received in a BGP REACH UPDATE. Receiving an unchanged UPDATE MUST NOT trigger this event.

6. LOST_ES: A BGP UNREACH UPDATE for a previously received Ethernet Segment route has been received. If an UNREACH is seen for a route that has not been advertised previously, the event MUST NOT be triggered.

7. CALCULATED: DF has been succesfully calculated.

According actions when transitions are performed or states entered/ exited:

1. ANY STATE on ES_DOWN: (i)stop DF timer (ii) assume non-DF for local PE

2. INIT on ES_UP: (i)do nothing

3. INIT on RCVD_ES, LOST_ES: (i)do nothing

4. DF_WAIT on entering the state: (i) start DF timer if not started already or expired (ii) assume non-DF for local PE

5. DF_WAIT on RCVD_ES, LOST_ES: do nothing

6. DF_WAIT on DF_TIMER: do nothing

7. DF_CALC on entering or re-entering the state: (i) rebuild according list and hashes and perform election (ii) FSM generates CALCULATED event against itself

8. DF_CALC on LOST_ES or VLAN_CHANGE: do nothing

9. DF_CALC on RCVD_ES: do nothing

10. DF_CALC on CALCULATED: (i) mark election result for VLAN or bundle

11. DF_DONE on exiting the state: (i)if RFC7432 election or new election and lost primary DF then assume non-DF for local PE for VLAN or VLAN bundle.

12. DF_DONE on VLAN_CHANGE or LOST_ES: do nothing

8. Operational Considerations

TBD.

9. Security Considerations

This document raises no new security issues for EVPN.

10. Acknowledgements

The authors would like to thank Tamas Mondal, Sami Boutros, Jakob Heitz, Jorge Rabadan and Patrice Brissette for useful feedback and discussions.

11. References

11.1. Normative References

[HRW1999] Thaler, D. and C. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates", IEEE/ACM Transactions in networking Volume 6 Issue 1, February 1998.

[I-D.ietf-idr-extcomm-iana] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", draft-ietf-idr-extcomm-iana-02 (work in progress), December 2013.

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

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <http://www.rfc-editor.org/info/rfc4360>.

[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>.

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

11.2. Informative References

[CHASH] Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M., and D. Lewin, "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web", ACM Symposium on Theory of Computing ACM Press New York, May 1997.

[CLRS2009] Cormen, T., Leiserson, C., Rivest, R., and C. Stein, "Introduction to Algorithms (3rd ed.)", MIT Press and McGraw-Hill ISBN 0-262-03384-4., February 2009.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <http://www.rfc-editor.org/info/rfc2991>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <http://www.rfc-editor.org/info/rfc2992>.

[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>.

[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>.

[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>.

Authors' Addresses

Satya Ranjan Mohanty
Cisco Systems, Inc.
225 West Tasman Drive
San Jose, CA 95134
USA

Email: satyamoh@cisco.com

Keyur Patel
Cisco Systems, Inc.
225 West Tasman Drive
San Jose, CA 95134
USA

Email: keyupate@cisco.com

Ali Sajassi
Cisco Systems, Inc.
225 West Tasman Drive
San Jose, CA 95134
USA

Email: sajassi@cisco.com

John Drake
Juniper Networks, Inc.
1194 N. Mathilda Drive
Sunnyvale, CA 95134
USA

Email: jdrake@juniper.com

Antoni Przygienda
Ericsson
300 Holger Way
San Jose, CA 95134
USA

Email: antoni.przygienda@ericsson.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-mld-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/1id-abstracts.html

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

Table of Contents

1 Introduction 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 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.

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:

- only hosts/VMs
- mix of hosts/VMs and multicast source
- mix of hosts/VMs, multicast source, and multicast router

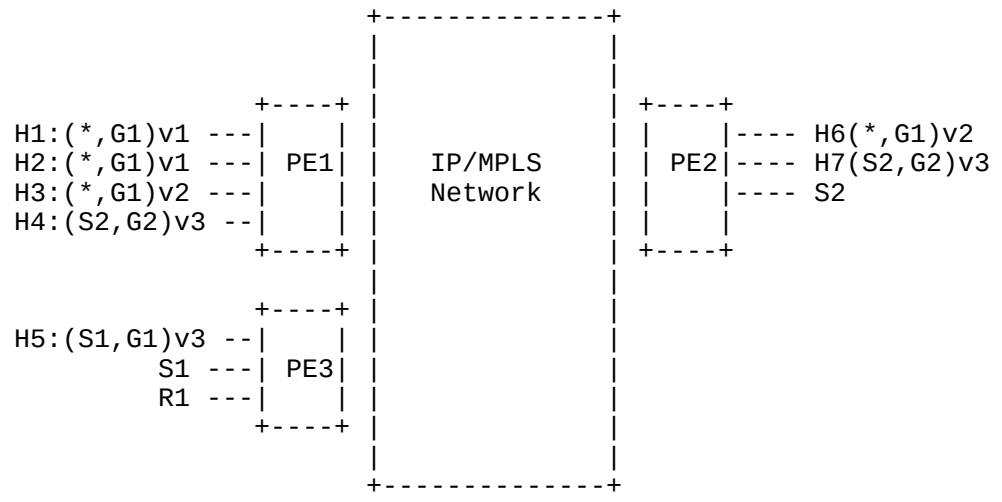


Figure 1:

3.1 PE with only attached hosts/VMs for a given subnet

When PE1 receives an IGMPv1 Join Report from H1, it does not forward this join to any of its other ports (for this subnet) because all these local ports are associated with the hosts/VMs. PE1 sends an EVPN Multicast Group route corresponding to this join for (*,G1) and setting v1 flag. This EVPN route is received by PE2 and PE3 that are the member of the same EVI. PE3 reconstructs IGMPv1 Join Report from this EVPN BGP route and only sends it to the port(s) with multicast routers attached to it (for that subnet). In this example, PE3 sends the reconstructed IGMPv1 Join Report for (*,G1) to only R1. Furthermore, PE2 although receives the EVPN BGP route, it does not 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.

3.1 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

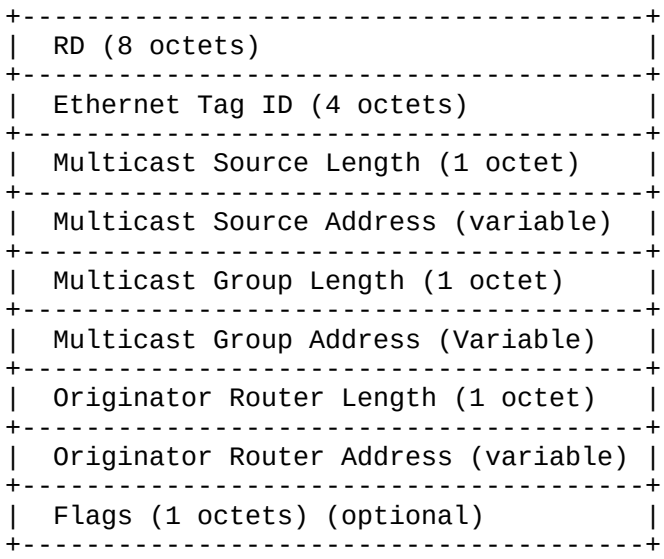
This document defines a new BGP EVPN route to carry IGMP membership reports. This route type is known as:

+ 6 - Selective Multicast Ethernet Tag Route

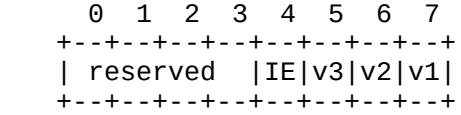
The detailed encoding and procedures for this route type is described in subsequent section.

5.1 Selective Multicast Ethernet Tag Route

An Selective Multicast Ethernet Tag route type specific EVPN NLRI consists of the following:



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:



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 forth 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 Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

Ali Sajassi
Samer Salam
Dennis Cai
Cisco

Expires: April 19, 2016

October 19, 2015

Multi-homed L3VPN Service with Single IP peer to CE
draft-sajassi-bess-evpn-l3vpn-multihoming-00

Abstract

This document describes how EVPN can be used to offer a multi-homed L3VPN service leveraging EVPN Layer 2 access redundancy. The solution offers a single IP peer to the Customer Edge (CE) nodes, rapid failure detection, minimal fail-over time and make-before-break paradigm for maintenance.

Table of Contents

1 Introduction 3
1.1 Terminology 3
2 Requirements 3
3 Challenges with L3VPN Multi-homing 4
4 Solution 5
5 Failure Scenarios 6
5.1 Pseudowire Failure 6
5.2 PE Node Failure 7
6 Security Considerations 7
7 IANA Considerations 7
8 References 7
8.1 Normative References 7
8.2 Informative References 7
Authors' Addresses 7

1 Introduction

[RFC7432] defines EVPN, a solution for multipoint Layer 2 Virtual Private Network (L2VPN) services, with advanced multi-homing capabilities, using BGP for distributing customer/client MAC address reachability information over the core MPLS/IP network. [EVPN-IRB] and [EVPN-PREFIX] discuss how EVPN can be used to support inter-subnet forwarding among hosts across different IP subnets, while maintaining the redundancy capabilities of the original solution.

In this document, we discuss how EVPN can be used to offer a multi-homed L3VPN service leveraging its Layer 2 access redundancy. The solution offers a single IP peer to the Customer Edge (CE) nodes, rapid failure detection, minimal fail-over time and make-before-break paradigm for maintenance.

1.1 Terminology

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

2 Requirements

The network topology in question comprises of three domains: the customer network, the MPLS access network and the MPLS core network, as shown in the figure below.

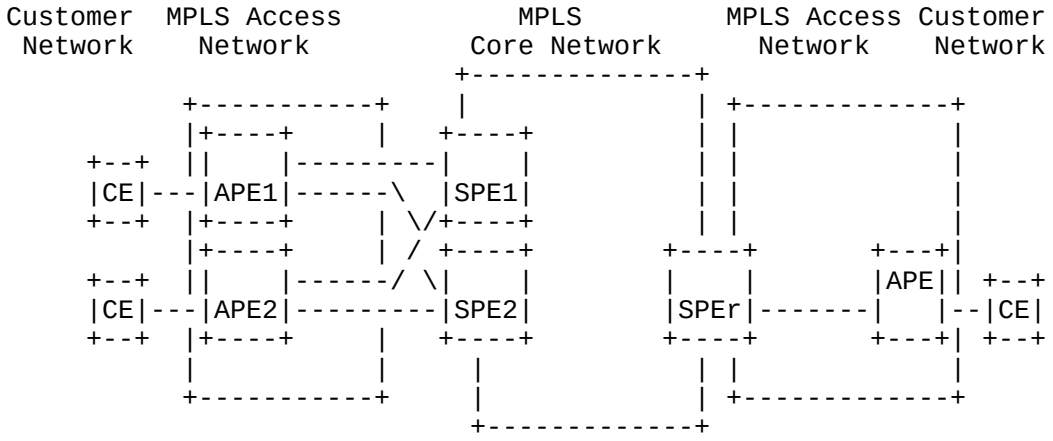


Figure 1: Network Topology

The customer network connects via Customer Edge (CE) nodes to the

MPLS Access Network. The MPLS Access Network includes Access PEs (A-PEs) and MPLS P nodes (not shown for simplicity). The A-PEs provide a Virtual Private Wire Service (VPWS) to the connected CEs using Ethernet over MPLS (EoMPLS) pseudowires per [RFC5462]. The access pseudowires terminate on the service PEs (S-PE1, S-PE2,..., S-PEr). The Service PEs (S-PEs) provide inter-subnet forwarding between the CEs, i.e. L3VPN service between them. To provide redundancy, pseudowires from a given A-PE can terminate on two or more S-PEs forming a Redundancy Group. This provide multi-homed interconnect of A-PEs to S-PEs.

The solution MUST support the following requirements:

- The S-PEs in a redundancy group must provide single-active redundancy to the CEs, i.e. only one S-PE is actively forwarding traffic at any given point of time.

- The SPEs in a redundancy group must appear as a single IP peer to the CE, and a single eBGP session will be established between a given CE and its associated S-PEs.

- In the case of S-PE failure, pseudowire failure or S-PE isolation from access network, the fail-over time should be minimized by optimizing both the backup pseudowire establishment as well as the BGP convergence time. This reduces the amount of traffic loss as the active path reroutes to one of the backup S-PEs.

- The active S-PE must be able to quickly detect pseudowire failures or its isolation from the access MPLS network by means of a proactive monitoring mechanism.

- For system maintenance, it should be possible to support a make-before-break paradigm, where the backup path is in warm standby state before a given active S-PE is taken offline for service.

3 Challenges with L3VPN Multi-homing

The requirements depicted in section 2 above, especially the requirement to maintain a single eBGP session between the CE and the S-PEs, introduce challenges for standard L3VPN multi-homing solutions. In particular, the BGP prefix independent convergence (PIC) solution [BGP-PIC] cannot be used here because the backup S-PEs have no means of learning the IP prefixes from the CE: recall that the CE will only have an active eBGP session with the active S-PE. As a result, when the primary S-PE fails, the backup S-PE will have no alternate paths to the prefixes advertised by the CE. Therefore, with BGP PIC it is not possible to address the fast fail-over requirement.

4 Solution

The solution involves running EVPN on the S-PEs in single-active redundancy mode albeit for inter-subnet forwarding (i.e. Layer 3 forwarding). All pseudowires associated with a given CE are considered collectively as a Virtual Ethernet Segment (vES) [Virtual-ES] from the EVPN PEs perspective.

In the MPLS access network, pseudowire redundancy mechanisms are used [RFC6718][RFC6870] in either the Independent mode or the Master/Slave mode, with the S-PEs acting as the Master. The EVPN Designated Forwarder (DF) election mechanism is used to identify the active and standby S-PEs, and the pseudowire Preferential Forwarding Status Bit [RFC6870], for the access pseudowires, is derived from the outcome of the DF election, as follows:

- The S-PE that is elected as DF for a given vES MUST advertise Active in the Preferential Forwarding Status bit over the pseudowire corresponding to the vES.

- The SPE that is elected as non-DF for a given vES MUST advertise Standby in the Preferential Forwarding Status bit over the pseudowire corresponding to the vES.

On the S-PEs, the pseudowires from the Access PEs are terminated onto VRFs, such that all pseudowires within a given redundancy set terminate on a single IP endpoint on the S-PEs. To achieve this, the S-PEs in a given Redundancy Group are configured with the same Anycast IP and MAC addresses on the virtual (sub)interface corresponding to the VRF termination point.

Since the S-PEs are running in EVPN single-active redundancy mode, the S-PEs would advertise an Ethernet AD route per vES with the single-active flag set per [RFC7432]. Since only the DF S-PE has its access pseudowire in Active state, only that device would establish an eBGP session with the CE and receive control and data traffic. The DF S-PE advertises host prefixes that it receives, from the CE over the eBGP session, to other PEs in the EVI using EVPN route type-5, with the proper ESI set. Remote PEs learn the host prefixes and associate them with the ESI, using the advertising PE as the next-hop for forwarding.

Other S-PEs in the same Redundancy Group as the advertising PE will receive the same EVPN route type-5 advertisement, and will recognize the associated ESI as a locally attached vES. This information will be used in the case of failure to provide a backup path to the CE. In other words, the S-PEs in the same Redundancy Group, use EVPN Aliasing procedure to synchronzie their IP-VRFs among themselves. It

is worth noting here that the S-PEs in the Redundancy Group will have their ARP caches synchronized through the EVPN route type-2 advertisements from the DF PE.

5 Failure Scenarios

5.1 Pseudowire Failure

The active (DF) S-PE can proactively monitor the health of the primary pseudowire by using a pseudowire OAM mechanism such as VCCV-BFD. As such, the S-PE can detect the failure of the primary pseudowire, and react by withdrawing both the Ethernet Segment route as well as the Ethernet A-D route associated with the vES. Note that the S-PE advertises the Ethernet A-D route per vES granularity as well as the Ethernet A-D per EVI. The withdrawal of the Ethernet Segment route serves as an indication to the backup S-PE to go active (i.e. act as a backup DF), and activate its pseudowires to the Access PE. The withdrawal of the Ethernet A-D route triggers a "mass withdraw" on the remote PEs: these PEs adjust their next-hop associated with the prefixes that were originally advertised by the failed PE to point to the "backup path" per [RFC7432]. This provides relatively fast convergence because only a single message per Ethernet Segment is required for the remote PEs to switch over to the backup path irrespective of how many prefixes were learnt from the CE over the pseudowire. Also, note that no synchronization of VRF or ARP tables is required between the primary S-PE and its backup S-PE during the fail-over, because these tables were populated ahead of time during the original EVPN route advertisements.

As a result of the pseudowire failure, the eBGP session between the CE and the original DF PE will time out. This will cause said S-PE to start a timer in order to defer withdrawing the EVPN type-5 and type-2 routes that it had advertised for the prefixes learnt over the session from the CE. As the backup pseudowire to the backup DF PE goes active, the eBGP session will be re-established by the CE with the backup PE. Since both PEs share the same Anycast IP and MAC addresses, the CE does not recognize that it is in communication with a different PE.

To minimize disruption in data forwarding on the CE and the backup PE, the non-stop forwarding feature such as BGP Graceful Restart is used. Since the end-point IP address has not changed, this eBGP session handover between the primary S-PE and the backup S-PE, looks like a eBGP session flap with respect to the CE. Thus, the CE continues its packet forwarding operation in data-plane while synchronizing its control-plane with the backup S-PE.

5.2 PE Node Failure

In the case of PE node failure, the operation is similar to the steps described above, albeit that EVPN route withdrawals are performed by the Route Reflector instead of the PE.

6 Security Considerations

TBD.

7 IANA Considerations

TBD

8 References

8.1 Normative References

[RFC7432] Sajassi et al., "Ethernet VPN", RFC 7432, February 2015.

[EVPN-IRB] Sajassi et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-00, work in progress, November 2014.

[EVPN-PREFIX] Rabadan et al., "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-02, work in progress, September 2015.

[RFC6718] Muley P., et al., "Pseudowire Redundancy", RFC 6718, August 2012.

[RFC6870] Muley P., et al., "Pseudowire Preferential Forwarding Status Bit", RFC 6870, February 2013.

8.2 Informative References

[BGP-PIC] Bashandy A. et al., "BGP Prefix Independent Convergence", draft-rtgwg-bgp-pic-02.txt, work in progress, October 2013.

Authors' Addresses

Ali Sajassi
Cisco
EMail: sajassi@cisco.com

Samer Salam
Cisco
EMail: ssalam@cisco.com

Dennis Cai
Cisco
EMail: dcai@cisco.com

BESS Workgroup
Internet Draft
Intended Status: Standards Track

Ali Sajassi
Samer Salam
Cisco
Luay Jalil
Verizon
John Drake
Tapraj Singh
Juniper

Expires: April 19, 2016 October 19, 2015

PBB-EVPN with Anycast IP Tunnels
draft-sajassi-bess-pbb-evpn-anycast-ip-tunnels-00

Abstract

This document describes how PBB-EVPN can be combined with Anycast IP Tunnels to provide resilient Layer 2 services with simplified operations on Access Nodes (ANs).

Status of this Memo

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

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

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

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

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

Table of Contents

1 Introduction 3
1.1 Terminology 3
2. Requirements 4
3 Current Challenges with PBB-EVPN 4
3.1 All-Active Redundancy Mode 4
3.2 Single-Active Redundancy Mode 5
4 Solution 5
4.1 Overview 5
4.2 Operation 5
4.2.1 Known Unicast Traffic 6
4.2.2 BUM Traffic 7
4.3 Optimizing PE B-MAC Address Allocation 8
4.3.1 Known Unicast Traffic 8
4.3.2 BUM Traffic 8
4.3.2.1 BUM Traffic from Ethernet Segment 8
4.3.2.2 BUM Traffic from PE in Redundancy Group 8
4.3.2.3 BUM Traffic from Remote PE 8
5 Failure Scenarios 9
5.1 Link/Node Failure in Access Network 9
5.2 PE Node Failure 9
5.3 IP Tunnel Failure 9
5.4 AN Node Failure 10
6 Security Considerations 10
7 IANA Considerations 10
8 References 10
8.1 Normative References 10
8.2 Informative References 10
Authors' Addresses 11

1 Introduction

RFC7623 defines PBB-EVPN, a solution that provides scalable MPLS Layer 2 VPN services using multi-protocol BGP combined with Provider Backbone Bridging (PBB) [802.1ah].

In this document, we describe a solution that uses PBB-EVPN to aggregate IP access networks over an MPLS backbone, while offering Layer 2 VPN services end to end. The network topology in question comprises of three domains: the customer network, the IP access network and the MPLS backbone, as shown in the figure below.

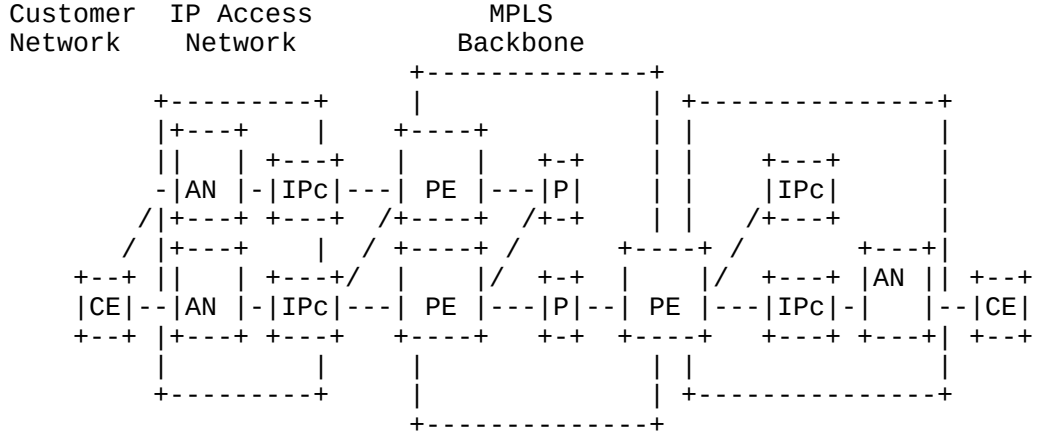


Figure 1: Target Topology

The customer network connects via Customer Edge (CE) devices to the IP Access Network. The IP Access Network includes Access Nodes (ANs) and IP core nodes (IPc). The ANs perform tunneling of Ethernet packets using some IP tunneling mechanism (e.g. GRE or VXLAN). The MPLS Backbone comprises of PBB-EVPN PEs as well as MPLS core nodes (P). The PBB-EVPN PEs terminate the IP tunnels which originate from the ANs in their local IP Access Network.

To simplify the operations and reduce the provisioning overhead on the ANs, as well as to provide resiliency, the PEs will use Anycast IP addresses as the tunnel destination, for tunnels originating from the ANs. We will refer to this setup as PBB-EVPN with Anycast IP tunnels.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in RFC 2119 [RFC2119].

2. Requirements

The solution MUST support multipoint Layer 2 VPN services end to end (i.e. CE to CE), with the following requirements:

- Support for IP as well as non-IP payloads. Hence, the solution should not rely on any ARP or ND snooping mechanism, but rather rely on MAC learning in data-plane.

- Use IP as the underlying transport in the access network, with support for IPv6.

- Support VLAN multiplexing with Customer VLAN (C-Tag) transparency in the IP tunnels.

- Support for VLAN aware service bundling over the IP tunnels on the PBB-EVPN PEs. This means that the PE needs to identify the L2 Bridge Domain based on the combination of the IP tunnel identifier AND the C-Tag (and/or S-tag).

- Support for local switching between IP tunnels on the PBB-EVPN PEs. This is due to the assumption that the ANs may not support any local switching/bridging functionality between their access ports.

- Support for hierarchical QoS with two levels in the hierarchy: per IP tunnel and per C-Tag (and/or S-tag).

- Provide resilient interconnect with protection against: PE node failure, path failures in the IP access network, and IP tunnel failure.

- Simplify the provisioning of the ANs by using Anycast IP addresses as the tunnel destination on the PEs. This eliminates the need to explicitly provisioning the ANs with the unicast IP addresses of the redundant PEs.

3 Current Challenges with PBB-EVPN

PBB-EVPN currently can operate in two different redundancy modes: All-Active redundancy and Single-Active redundancy. Neither of these modes can satisfy the requirements set forth in section 2 above. We will discuss this in detail in this section.

3.1 All-Active Redundancy Mode

The challenge with the All-Active redundancy mode is that the traffic arriving from the MPLS backbone get load-balanced among the PEs in the redundancy group. As such, it is not possible to enforce the QoS policies reliably for traffic in the backbone to access direction. Note that one cannot assume that the traffic will get distributed evenly between the PEs due to the possibility of "elephant" flows (i.e. flows with considerable bandwidth demands), especially when load-balancing algorithms on the PEs do not take bandwidth into account in the hashing decision.

3.2 Single-Active Redundancy Mode

The challenge with the Single-Active redundancy mode is that, based on the DF election, only one of the PBB-EVPN PEs will be forwarding traffic from access to the backbone. At the same time, depending on the IGP distance between the Access Node (AN), and the PEs, traffic over the Anycast IP tunnel may be delivered to the non-DF PE, where it is permanently dropped. This is because the DF election procedures, which are triggered by BGP exchanges, are independent of the IGP path calculations.

Relaxing the DF filtering rules will result in duplicate packets, and hence is not an option.

4 Solution

4.1 Overview

The solution involves defining a new asymmetric redundancy scheme for PBB-EVPN, which behaves similar to All-Active redundancy for traffic destined to the MPLS backbone from the Ethernet Segment and behaves similar to Single-Active redundancy for traffic destined to the Ethernet Segment from the MPLS backbone.

Since the mechanism behaves like All-Active redundancy for traffic destined towards the MPLS backbone, the PEs in the redundancy group can all receive traffic from the ANs, irrespective of how the IGP steers the Anycast traffic. The segment split horizon filtering on the PE prevents looping of BUM traffic.

Moreover, since the mechanism behaves like Single-Active redundancy for traffic destined towards the Ethernet Segment, the active PE can reliably enforce the QoS policies.

4.2 Operation

Each IP tunnel will be treated as a virtual Ethernet Segment (vES) [Virtual-ES] on the PBB-EVPN PEs. However, instead of having a single

B-MAC address associated with each vES across all PEs in the RG, each PE will assign a different B-MAC for each vES, because Single-Active redundancy is used for traffic from the MPLS backbone. This ensures that remote PEs learn the C-MAC addresses against the unique B-MAC of the active PE only. Note that the active PE here is actually determined based on where the Anycast traffic lands, according the IGP distance between the ANs and the PEs in the IP access network.

Traffic filtering based on DF election applies only to BUM traffic, and only for traffic in the direction towards the Ethernet Segment.

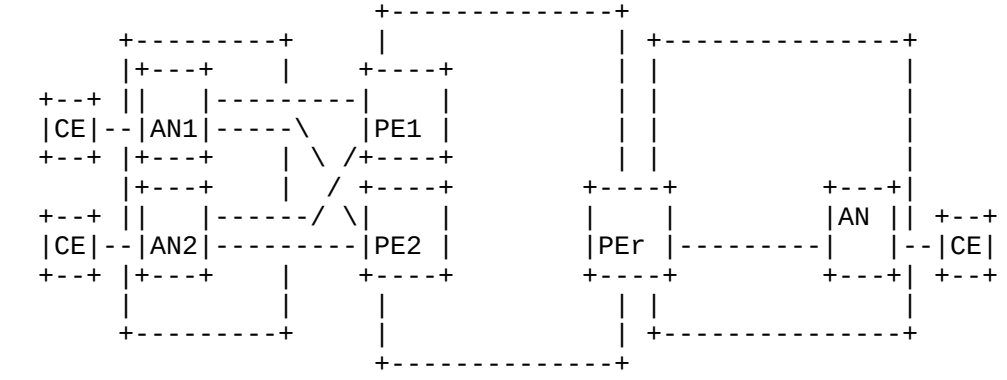


Figure 2: Example Network

For what follows, please refer to the example network in Figure 2 above.

4.2.1 Known Unicast Traffic

When an access nodes (e.g. AN1) forwards known unicast traffic over the IP access network, the traffic will be steered towards one of the PEs depending on the IGP distance between the AN and the PEs. In the case where both PEs are equidistant from the AN, i.e. ECMP, the load-balancing hash on the AN and IP core nodes of the access network determines which PE receives the traffic. Assume the traffic is delivered to PE1 in this example. PE1 would then encapsulate the traffic in the PBB header, using its B-MAC address that is associated with the vES corresponding to the IP tunnel on which the traffic was received. The PE then adds the MPLS encapsulation and forwards the packets towards the destination remote PE (PEr in this example). PEr would then learn the C-MAC against the B-MAC address associated with PE1. Hence, for the reverse traffic PEr will forward the traffic destined to that C-MAC to PE1. This follows normal PBB-EVPN operation.

The only issue to consider is how would the PEs enforce the QoS policies in case of the presence of multiple equal cost paths between the AN and the PEs. Since the QoS policies need to be enforced per IP tunnel and per C-Tag (and/or S-tag) on the PE, the AN needs to ensure that all traffic of a given tunnel lands on the same PE. This can be done by encoding the tunnel identifier in the entropy field of the packet. For example, in the case of VXLAN, the source VTEP address can be hashed into the source UDP port. Similarly, in the case of GRE, the source tunnel address can be hashed into the GRE key. This guarantees that all traffic associated with a given IP tunnel from an AN is destined to a single PE, even in the presence of ECMP.

4.2.2 BUM Traffic

The PEs in a redundancy group perform the standard DF election procedures described in RFC7623 (i.e. with per I-SID load-balancing). As a result, only one of the PEs will be responsible for forwarding BUM traffic received from the MPLS backbone towards the Ethernet Segment (i.e. IP access network). This is per standard PBB-EVPN operation.

For BUM traffic from the IP access network destined towards the MPLS backbone, the traffic will be sent as a unicast from the AN to one of the PEs (depending on the IGP distance, or the ECMP hash in case of equidistant PEs). This is because the AN does not support any bridging function. The ingress PE, which receives the BUM traffic, will flood it following the standard procedures of PBB-EVPN. This includes applying any DF filtering on locally attached Ethernet Segments. For example, assume that for a given I-SID (ISID1), PE1 is the DF for the vES associated with the IP tunnel to AN1, whereas PE2 is the DF for the vES associated with the IP tunnel to AN2. For BUM traffic sent by AN1 to PE1, the PE1 will not flood the traffic towards AN2 since this PE is not the DF for that vES.

The ingress PE would flood the BUM traffic to other PEs in the redundancy group. In our example, this means that PE1 will flood the BUM traffic to PE2. These PEs need to apply the segment split-horizon filtering function to guarantee that the BUM traffic does not loop back into the originating AN. To achieve this, a new procedure is added to standard PBB-EVPN whereby the PE examines the source B-MAC address on the BUM traffic, and recognizes that this B-MAC is associated with a local vES. As such, the PE does not flood the BUM traffic over that specific vES. This is effectively an extension of the PBB-EVPN split-horizon functionality: in the standard, a single B-MAC is associated with a given ESI, whereas here multiple B-MACs are associated with the same ESI. Note that the PE would forward the BUM traffic on other virtual Ethernet Segments in the same bridge domain, and for which it is the assigned DF. Going back to our

example, PE2 in this case would forward the BUM traffic received from PE1 towards AN2 but not towards AN1.

4.3 Optimizing PE B-MAC Address Allocation

The mechanisms described thus far require the allocation of a B-MAC address per IP tunnel (i.e. per CE). For high-scale deployments, this allocation scheme defeats the scaling benefits of PBB-EVPN. Hence, a more optimal B-MAC address allocation mechanism is needed. This section describes how this can be achieved.

In order to reduce the number of B-MAC addresses required per PE, the approach is to adopt the "local bias" mechanism defined in [Overlay] for PBB-EVPN. This allows only a single B-MAC address to be assigned to each PE for all its vES’s (instead of one B-MAC per vES).

4.3.1 Known Unicast Traffic

The operation for known unicast traffic is similar to what was described in section 4.2.1, albeit with a single source B-MAC address being used for all traffic arriving at the PE from the virtual Ethernet Segments associated with the IP tunnels.

4.3.2 BUM Traffic

The operation for BUM traffic can be broken down to three different scenarios, as discussed next.

4.3.2.1 BUM Traffic from Ethernet Segment

For BUM traffic arriving from the (virtual) Ethernet Segment, the PE follows the local bias mechanisms. That is, it floods the traffic over all other local (virtual) Ethernet Segments in the same bridge-domain irrespective of DF election state. The PE also floods the traffic over the MPLS backbone.

4.3.2.2 BUM Traffic from PE in Redundancy Group

For BUM traffic arriving from another ingress PE in the same Redundancy Group, the PE inspects the source B-MAC address in the packets to identify the right replication list for the I-SID in question. This replication list would exclude all (virtual) Ethernet Segments that are common with the ingress PE (to prevent loops). The PE would flood the BUM traffic over all (virtual) Ethernet Segments in that specific replication list, subject to the DF filtering constraints - i.e., it is the DF for that segment.

4.3.2.3 BUM Traffic from Remote PE

For BUM traffic arriving from a remote ingress PE that has no segments in common with the egress PE, the latter would again examine the source B-MAC address in the packets to identify the replication list for the I-SID in question. This replication list would include all (virtual) Ethernet Segments that are in the associated bridge-domain. The PE would flood the traffic over all those segments while observing the DF filtering rules.

5 Failure Scenarios

In this section we will discuss the operation of the solution in case of various failure scenarios.

5.1 Link/Node Failure in Access Network

For link or node failures in the access network, which do not cause the ANs to completely lose connectivity to the PEs, the failure will trigger the IGP to recalculate the shortest path. This may cause the Anycast traffic from some ANs to be steered towards new PEs. However, no action will be required in the control plane on the PEs. Remote PEs will automatically update their C-MAC/B-MAC associations through data-plane learning. There are no transient loops, and the convergence time is a function of how quickly the IGP can re-converge.

5.2 PE Node Failure

Failure of the PE node is addressed by the fact that the IP tunnels use Anycast addresses. The ANs do not need to explicitly react to the failure in any special way, as the IGP re-convergence takes care of the fail-over procedures. The B-MAC routes advertised by the failed PE are withdrawn by the Route Reflector, which in turn flushes all the C-MACs associated with those B-MACs on the remote PEs. If this happens before any new traffic arrives from the CE via the new PE, temporary flooding would occur as expected. However, as soon as those remote PEs start receiving traffic from the associated C-MACs over the new PE, the remote PEs will update their C-MAC/B-MAC bindings through normal data-plane learning.

5.3 IP Tunnel Failure

IP tunnel failure occurs when there is no viable path, within the access network, between the AN and a PE. The PE can detect such condition by using IGP route watch mechanisms. Upon detecting this failure, the PE reacts with two actions:

- It withdraws all the Ethernet Segment routes associated with the unreachable AN.

- It withdraws all the MAC Advertisement routes associated with the B-MAC(s) of the affected vES's.

Similar to the PE node failure scenario above, this may result in temporary flooding from remote PEs until traffic starts flowing in the other direction from the CEs.

5.4 AN Node Failure

From the perspective of the PE, this failure is handled the same way as the IP tunnel failure scenario. If the CE connected to the failed AN is single-homed, then no redundancy would be available.

6 Security Considerations

There are no special security considerations beyond those of PBB-EVPN.

7 IANA Considerations

TBD

8 References

8.1 Normative References

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

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

[PBB] IEEE, "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", Clauses 25 and 26, IEEE Std 802.1Q, DOI 10.1109/IEEESTD.2011.6009146.

[Overlay] Sajassi et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01, work in progress, February 2015.

8.2 Informative References

Authors' Addresses

Ali Sajassi
Cisco
EMail: sajassi@cisco.com

Samer Salam
Cisco
EMail: ssalam@cisco.com

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

John Drake
Juniper
EMail: jdrake@juniper.net

Tapraj Singh
Juniper
EMail: tsingh@juniper.net

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2016

H. Shah
Ciena Corporation
P. Brissette
R. Rahman
K. Raza
Cisco Systems, Inc.
Z. Li
Z. Shunwan
W. Haibo
Huawei Technologies
I. Chen
Ericsson
M. Bocci
Alcatel-Lucent
J. Hardwick
Metaswitch
S. Esale
K. Tiruveedhula
T. Singh
Juniper Networks
I. Hussain
Infinera Corporation
B. Wen
J. Walker
Comcast
N. Delregno
L. Jalil
M. Joecylyn
Verizon

October 15, 2015

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

Abstract

This document describes a YANG data model for Layer 2 VPN services over MPLS networks. These services include Virtual Private Wire Service (VPWS) and Virtual Private LAN service (VPLS) that uses LDP and BGP signaled Pseudowires. The current version of the document expands the L2VPN object model to include VPLS services in addition to the VPWS services described in the last revision. This is a living document and contains aspects of object models that have been discussed extensively in the working group with consensus. The intention is to continue to seek input from larger audience during evolution of the L2VPN service model through this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Specification of Requirements 4
3. L2VPN YANG Model 4
3.1. Overview 4
3.2. L2VPN Common 7
3.2.1. ac-templates 7
3.2.2. pw-templates 7
3.3. VPWS and VPLS 8
3.3.1. ac list 8
3.3.2. pw list 8
3.3.3. redundancy-grp choice 8
3.3.4. endpoint container 8
3.3.5. vpws-instances and vpls-instances container 9
4. YANG Module 13

5. Security Considerations 30
6. IANA Considerations 30
7. Acknowledgments 30
8. References 30
8.1. Normative References 30
8.2. Informative References 30
Authors' Addresses 33

1. Introduction

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

This document introduces a YANG data model for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] as well as switching between the local attachment circuits. The L2VPN services include point-to-point VPWS and Multipoint VPLS services. These services are realized by signaling Pseudowires across MPLS networks using LDP [RFC4447][RFC4762] or BGP[RFC4761].

The Yang data model in this document defines Ethernet based Layer 2 services. Other Layer 2 services, such as ATM, Frame Relay, TDM, etc are included in the scope but will be covered as the future work items. The Ethernet based Layer 2 services will leverage the definitions used in other standards organizations such as IEEE 802.1 and Metro Ethernet Forum (MEF).

The goal is to propose a data object model consisting of building blocks that can be assembled in different order to realize different services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The VPWS service definitions were covered first in the last revision of the document. The current version documents VPLS services that build on the data blocks defined for VPWS.

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

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

The document is organized to first define the data model for the configuration of all the L2VPN services followed by definition of operational state, actions and notifications for the same. The L2VPN data object model defined in this document uses the instance centric approach. The attributes of each service, VPWS, VPLS, etc are specified for a given service instance.

2. 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, mpls-l2vpn, is defined as a parent for three different second level containers that are vpws-instances, vpls-instances, and common building blocks of AC-templates(Attachment Circuit templates) and pseudowire-templates. The current version of the document is extended to include vpls-instances.

The L2VPN services have been defined in the IETF L2VPN working group but leverages the pseudowire technologies that were defined in the PWE3 working group. A large number of RFCs from these working groups cover this subject matter. Hence, it is prudent that this document state the scope of the MPLS L2VPN object model definitions.

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

- Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC4447]
- Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]

- Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- Segmented Pseudowire [RFC6073]
- Framework for Layer 2 Virtual Private Networks [RFC4664]
- Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]
- Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]
- Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- Pseudowire status for static pseudowire [RFC6478]

Note that while pseudowire over MPLS-TP related work is in scope, the initial effort will only address definitions of object models for services that are commonly deployed.

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

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

```
template-ref AC // AC
                   template
                   attributes

template-ref PW // PW
                   template
                   attributes

vpls-instance name // container

        common attributes

        PBB-parameters // container
                          pbb specific attributes

        BGP-parameters // container
                          common attributes
                          auto-discovery attributes
                          signaling attributes

        // list of ACs and PWs being used
        AC // container
                template-ref AC
                attribute-override
        PW // container
                template-ref PW
                attribute-override

        // List of endpoints, where each member endpoint container is -
                AC // reference
                        PW // reference
                redundancy-grp // container
                        AC // reference
                        PW // reference

vpws-instance name // container

        common attributes

        BGP-parameters // container
                          common attributes
```

```
                        auto-discovery attributes
                        signaling attributes

        AC-1 // container
                template-ref AC
                attribute-override
        PW-2 // container
                template-ref PW
                attribute-override
        PW-3 // container
                template-ref PW
                attribute-override

        // ONLY 2 endpoints!!!
        endpoint-A // container
                        AC-1 // reference

        endpoint-Z // container
                redundancy-grp // container
                        PW-2 // reference
                        PW-3 // reference
```

Figure 1

3.2. L2VPN Common

3.2.1. ac-templates

The ac-templates container contains a list of ac-template. Each ac-template defines a list of AC attributes that are part of native services but associated and processed within the context of L2VPN. For instance, Ethernet VLAN tag imposition, disposition and translation or CVID-bundling would be part of this template. The ac-template defintion remains skeleton. More details will be supplemented from the external documents prepared by MEF and IEEE802.1

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. VPWS and VPLS

3.3.1. ac list

Each VPWS and VPLS instance defines a list of AC that are participating members of the given service instance. Each entry of the AC consists of one ac-template with predefined attributes and values, but also defines attributes that override the attributes defined in referenced ac-template. The VPLS specific attribute(s) are present in the definition of AC that are member of VPLS instance only and not applicable to VPWS service.

3.3.2. pw list

Each VPWS and VPLS instance defines a list of PW 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 -

- primary and backup AC
- primary and backup PW
- primary AC and backup PW
- 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 vpls-instances container

The vpws-instances container contains a list of vpws-instance. Each entry of the vpws-instance represents a layer-2 cross-connection of two endpoints. This model defines three possible types of endpoints, ac, pw, and redundancy-grp, and allows a vpws-instance to cross-connect any one type of endpoint to all other types of endpoint.

The vpls-instances container contains a list of vpls-instace. Each entry of the vpls-instance represent a list of endpoints that are member of the broadcast/bridge domain. The vpls-instance endpoints introduces an additional forwarding characteristics to a list of PWs and/or ACs. This split-horizon forwarding behavior is typical in VPLS instance.

The augmentation of ietf-mpls-l2vpn module is TBD. All IP addresses defined in this module are currently scoped under global VRF/table.

```
module: ietf-mpls-l2vpn
+--rw l2vpn
   +--rw common
   |  +--rw pw-templates
   |  |  +--rw pw-template* [name]
   |  |     +--rw name              string
   |  |     +--rw mtu?              uint32
   |  |     +--rw cw-negotiation?   cw-negotiation-type
   |  |     +--rw tunnel-policy?    string
   |  +--rw ac-templates
   |     +--rw ac-template* [name]
   |        +--rw name    string
   +--rw vpls-instances
   |  +--rw vpls-instance* [name]
   |     +--rw name              string
   |     +--rw mac-withdraw?     boolean
   |     +--rw pbb-parameters
   |     |  +--rw component-type?     pbb-component-type
   |     |  +--rw bind-b-component?   vpls-instance-ref
   |     |  +--rw i-tag?              uint32
```

```
|     |  +--rw backbone-src-mac?   yang:mac-address
|     +--rw bgp-parameters
|     |  +--rw common
|     |  |  +--rw route-distinguisher?   string
|     |  |  +--rw vpn-targets* [rt-value]
|     |  |     +--rw rt-value    string
|     |  |     +--rw rt-type     bgp-rt-type
|     |  +--rw discovery
|     |  |  +--rw vpn-id?   string
|     |  +--rw signaling
|     |     +--rw site-id?      uint16
|     |     +--rw site-range?   uint16
|     +--rw pw* [name]
|     |  +--rw name                    string
|     |  +--rw split-horizon-group?    string
|     |  +--rw template?               pw-template-ref
|     |  +--rw discovery-type?         l2vpn-discovery-type
|     |  +--rw signaling-type?         l2vpn-signaling-type
|     |  +--rw peer-ip?                inet:ip-address
|     |  +--rw pw-id?                  uint32
|     |  +--rw transmit-label?         uint32
|     |  +--rw receive-label?          uint32
|     +--rw ac* [name]
|     |  +--rw name                    string
|     |  +--rw split-horizon-group?    string
|     |  +--rw template?               ac-template-ref
|     +--rw endpoint* [id]
|        +--rw id                      uint8
|        +--rw split-horizon-group?    string
|        +--rw (ac-or-pw-or-redundancy-grp)?
|           +--:(ac)
|           |  +--rw ac* [name]
|           |     +--rw name    -> ../../../ac/name
|           +--:(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?            -> ../../ac/name
|              +--rw (backup)?
|              |  +--:(backup-pw)
|              |  |  +--rw backup-pw* [name]
|              |  |     +--rw name          -> ../../../pw/name
|              |  |     +--rw precedence?   uint32
```

```
   |              |  +--:(backup-ac)
   |              |     +--rw backup-ac?             -> ../../ac/name
   |              +--rw 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 service-type?     l2vpn-service-type
         +--rw discovery-type?   l2vpn-discovery-type
         +--rw signaling-type    l2vpn-signaling-type
         +--rw bgp-parameters
         |  +--rw common
         |  |  +--rw route-distinguisher?   string
         |  |  +--rw vpn-targets* [rt-value]
         |  |     +--rw rt-value    string
         |  |     +--rw rt-type     bgp-rt-type
         |  +--rw discovery
         |  |  +--rw vpn-id?   string
         |  +--rw signaling
         |     +--rw site-id?      uint16
         |     +--rw site-range?   uint16
         +--rw pw* [name]
         |  +--rw name              string
         |  +--rw cw-negotiation?   cw-negotiation-type
         |  +--rw template?         pw-template-ref
         |  +--rw vccv-ability?     boolean
         |  +--rw tunnel-policy?    string
         |  +--rw request-vlanid?   uint16
         |  +--rw vlan-tpid?        string
         |  +--rw ttl?              uint8
         |  +--rw (pw-type)?
         |     +--:(ldp-pw)
         |     |  +--rw peer-ip?          inet:ip-address
         |     |  +--rw pw-id?            uint32
         |     |  +--rw transmit-label?   uint32
         |     |  +--rw receive-label?    uint32
         |     |  +--rw icb?              boolean
         |     +--:(bgp-pw)
         |     |  +--rw remote-pe-id?     inet:ip-address
         |     +--:(bgp-ad-pw)
         |        +--rw remote-ve-id?     uint16
         +--rw ac* [name]
         |  +--rw name                       string
```

```
|  +--rw template?                  ac-template-ref
|  +--rw pipe-mode?                 enumeration
|  +--rw link-discovery-protocol?  link-discovery-protocol-type
+--rw endpoint-a
|  +--rw (ac-or-pw-or-redundancy-grp)?
|     +--:(ac)
|     |  +--rw ac?                -> ../../ac/name
|     +--:(pw)
|     |  +--rw pw?                -> ../../pw/name
|     +--:(redundancy-grp)
|        +--rw (primary)
|        |  +--:(primary-pw)
|        |  |  +--rw primary-pw?        -> ../../pw/name
|        |  +--:(primary-ac)
|        |     +--rw primary-ac?        -> ../../ac/name
|        +--rw (backup)
|        |  +--:(backup-pw)
|        |  |  +--rw backup-pw?         -> ../../pw/name
|        |  +--:(backup-ac)
|        |     +--rw backup-ac?         -> ../../ac/name
|        +--rw protection-mode?   enumeration
|        +--rw reroute-mode?      enumeration
|        +--rw reroute-delay?     uint16
|        +--rw dual-receive?      boolean
|        +--rw revert?            boolean
|        +--rw revert-delay?      uint16
+--rw endpoint-z
   +--rw (ac-or-pw-or-redundancy-grp)?
      +--:(ac)
      |  +--rw ac?                -> ../../ac/name
      +--:(pw)
      |  +--rw pw?                -> ../../pw/name
      +--:(redundancy-grp)
         +--rw (primary)
         |  +--:(primary-pw)
         |  |  +--rw primary-pw?        -> ../../pw/name
         |  +--:(primary-ac)
         |     +--rw primary-ac?        -> ../../ac/name
         +--rw (backup)
         |  +--:(backup-pw)
         |  |  +--rw backup-pw?         -> ../../pw/name
         |  +--:(backup-ac)
         |     +--rw backup-ac?         -> ../../ac/name
         +--rw protection-mode?   enumeration
         +--rw reroute-mode?      enumeration
         +--rw reroute-delay?     uint16
         +--rw dual-receive?      boolean
         +--rw revert?            boolean
```

```
                +--rw 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-mpls-l2vpn@2015-06-30.yang"
module ietf-mpls-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-l2vpn";
  prefix "l2vpn";

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

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

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

  revision "2015-10-15" {
    description "Second revision " +
                "  - Added container vpls-instances " +
                "  - Rearranged groupings and typedefs to be reused " +
                "    across vpls-instance and vpws-instances";
    reference "";
  }

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

  /* identities */

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

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

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

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

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

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

   /* typedefs */

   typedef l2vpn-service-type {
     type enumeration {
       enum ethernet {
         description "Ethernet service";
       }
       enum ATM {
         description "Asynchronous Transfer Mode";
       }
       enum FR {
         description "Frame-Relay";
       }
       enum TDM {
         description "Time Division Multiplexing";
       }
     }
     description "L2VPN service type";
   }

   typedef l2vpn-discovery-type {
     type enumeration {
       enum manual {
```

```
        description "Manual configuration";
      }
      enum bgp-ad {
        description "Border Gateway Protocol (BGP) auto-discovery";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP)";
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN discovery type";
  }

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

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

```
   typedef cw-negotiation-type {
     type enumeration {
       enum "non-preferred" {
         description "No preference for control-word";
       }
       enum "preferred" {
         description "Prefer to have control-word negotiation";
       }
     }
     description "control-word negotiation preference type";
   }

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

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

   typedef pw-template-ref {
     type leafref {
       path "/l2vpn/common/pw-templates/pw-template/name";
     }
     description "pw-template-ref";
   }

   typedef ac-template-ref {
     type leafref {
       path "/l2vpn/common/ac-templates/ac-template/name";
     }
     description "ac-tempalte-ref";
   }

   typedef vpls-instance-ref {
```

```
      type leafref {
        path "/l2vpn/vpls-instances/vpls-instance/name";
      }
      description "vpls-instance-ref";
    }

    /* groupings */

    grouping pbb-parameters-grp {
      description "PBB parameters grouping";
      container pbb-parameters {
        description "pbb-parameters";
        leaf component-type {
          type pbb-component-type;
          description "PBB component type";
        }
        leaf bind-b-component {
          when "../component-type = 'i-component'" {
            description "Only applies to an i-component";
          }
          type vpls-instance-ref;
          description "Reference to the associated b-component";
        }
        leaf i-tag {
          type uint32;
          description "i-tag";
        }
        leaf backbone-src-mac {
          type yang:mac-address;
          description "backbone-src-mac";
        }
      }
    }

    grouping bgp-parameters-grp {
      description "BGP parameters grouping";
      container bgp-parameters {
        description "Parameters for BGP";
        container common {
          when "../../discovery-type = 'bgp-ad'" {
            description "Check discovery type: " +
                        "Can only configure BGP discovery if " +
                        "discovery type is BGP-AD";
          }
          description "Common BGP parameters";
          leaf route-distinguisher {
            type string;
            description "BGP RD";
```

```
        }
        list vpn-targets {
          key rt-value;
          description "Route Targets";
          leaf rt-value {
            type string;
            description "Route-Target value";
          }
          leaf rt-type {
            type bgp-rt-type;
            mandatory true;
            description "Type of RT";
          }
        }
      }
      container discovery {
        when "../../discovery-type = 'bgp-ad'" {
          description "BGP parameters for discovery: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "BGP parameters for discovery";
        leaf vpn-id {
          type string;
          description "VPN ID";
        }
      }
      container signaling {
        when "../../signaling-type = 'bgp'" {
          description "Check signaling type: " +
                      "Can only configure BGP signaling if " +
                      "signaling type is BGP";
        }
        description "BGP parameters for signaling";
        leaf site-id {
          type uint16;
          description "Site ID";
        }
        leaf site-range {
          type uint16;
          description "Site Range";
        }
      }
    }
  }

  grouping pw-type-grp {
    description "pseudowire type grouping";
```

```
     choice pw-type {
       description "A choice of pseudowire type";
       case ldp-pw {
         leaf peer-ip {
           type inet:ip-address;
           description "peer IP address";
         }
         leaf pw-id {
           type uint32;
           description "pseudowire id";
         }
         leaf transmit-label {
           type uint32;
           description "transmit lable";
         }
         leaf receive-label {
           type uint32;
           description "receive label";
         }
         leaf icb {
           type boolean;
           description "inter-chassis backup";
         }
       }
       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 vpls-pw-list-grp {
     description "vpls-pw-list-grp";
     list pw {
       key "name";
       leaf name {
         type leafref {
           path "../../../pw/name";
         }
         description "name of pseudowire";
```

```
      }
      description "vpls pseudowire list";
    }
  }

  grouping vpls-ac-list-grp {
    description "vpls-ac-list-grp";
    list ac {
      key "name";
      leaf name {
        type leafref {
          path "../../../ac/name";
        }
        description "Reference to an attachment circuit";
      }
      description "vpls attachment circuit list";
    }
  }

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

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

  grouping vpls-endpoint-grp {
    description "A vpls endpoint";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        uses vpls-ac-list-grp;
        description "reference to attachment circuits";
      }
      case pw {
        uses vpls-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 leafref {
                path "../../ac/name";
              }
              description "Reference an attachment circuit";
            }
          }
        }
        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 leafref {
                path "../../ac/name";
              }
              description "Reference an attachment circuit";
            }
            description "backup-ac";
          }
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        leaf ac {
          type leafref {
            path "../../ac/name";
          }
          description "reference to an attachment circuit";
        }
      }
      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 leafref {
                  path "../../ac/name";
                }
                description "primary attachment circuit";
              }
            }
          }
          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 leafref {
                  path "../../ac/name";
                }
                description "backup attachment circuit";
              }
            }
          }
          uses redundancy-group-properties-grp;
        }
      }
    }

    /* We can define vpls-endpoing-grp that has the same structure as
     * vpws-endpoing-grp, but has more endpoint options.
     */

    /* L2VPN YANG Model */

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

```
            description "pw-template";
            leaf name {
              type string;
              description "name";
            }
            leaf mtu {
              type uint32;
              description "pseudowire mtu";
            }
            leaf cw-negotiation {
              type cw-negotiation-type;
              default "preferred";
              description
                "control-word negotiation preference";
            }
            leaf tunnel-policy {
              type string;
              description "tunnel policy name";
            }
          }
        }
        container ac-templates {
          description "attachment circuit templates";
          /* To be fleshed out in future revisions */
          list ac-template {
            key "name";
            description "ac-template";
            leaf name {
              type string;
              description "name";
            }
          }
        }
      }
      container vpls-instances {
        /* To be fleshed out in future revisions */
        description "vpls-instances";
        list vpls-instance {
          key "name";
          description "A VPLS instance";
          leaf name {
            type string;
            description "Name of a VPLS instance";
          }
          leaf mac-withdraw {
            type boolean;
            description "Withdraw MAC";
          }
```

```
        uses pbb-parameters-grp;
        uses bgp-parameters-grp;
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf discovery-type {
            type l2vpn-discovery-type;
            description "VPLS discovery type";
          }
          leaf signaling-type {
            type l2vpn-signaling-type;
            description "VPLS signaling type";
          }
          leaf peer-ip {
            type inet:ip-address;
            description "peer IP address";
          }
          leaf pw-id {
            type uint32;
            description "pseudowire id";
          }
          leaf transmit-label {
            type uint32;
            description "transmit lable";
          }
          leaf receive-label {
            type uint32;
            description "receive label";
          }
        }
        list ac {
          key "name";
          description "attachment circuit";
          leaf name {
            type string;
            description "name";
```

```
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          leaf template {
            type ac-template-ref;
            description "attachment circuit template";
          }
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          uses vpls-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 service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
```

```
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description "Override the control-word negotiation " +
                        "preference specified in the " +
                        "pseudowire template.";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf vccv-ability {
            type boolean;
            description "vccvability";
          }
          leaf tunnel-policy {
            type string;
            description "Used to override the tunnel policy name " +
                        "specified in the pseduowire template";
          }
          leaf request-vlanid {
            type uint16;
            description "request vlanid";
          }
          leaf vlan-tpid {
            type string;
            description "vlan tpid";
          }
          leaf ttl {
            type uint8;
            description "time-to-live";
          }
          uses pw-type-grp;
        }
        list ac {
```

```
            key "name";
            description "attachment circuit";
            leaf name {
              type string;
              description "name";
            }
            leaf template {
              type ac-template-ref;
              description "attachment circuit template";
            }
            leaf pipe-mode {
              type enumeration {
                enum "pipe" {
                  value 0;
                  description "regular pipe mode";
                }
                enum "short-pipe" {
                  value 1;
                  description "short pipe mode";
                }
                enum "uniform" {
                  value 2;
                  description "uniform pipe mode";
                }
              }
              description "pipe mode";
            }
            leaf link-discovery-protocol {
              type link-discovery-protocol-type;
              description "link discovery protocol";
            }
          }
          container endpoint-a {
            description "endpoint-a";
            uses vpws-endpoint-grp;
          }
          container endpoint-z {
            description "endpoint-z";
            uses vpws-endpoint-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 TBD for their useful comments.

8. References

8.1. Normative References

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

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Reshad Rahman
Cisco Systems, Inc.

Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.

Email: skraza@cisco.com

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies

Email: Zhuangshunwan@huawei.com

Wang Haibo
Huawei Technologies

Email: rainsword.wang@huawei.com

Ing-When Chen
Ericsson

Email: ing-wher.chen@ericsson.com

Mathew Bocci
Alcatel-Lucent

Email: mathew.bocci@alcatel-lucent.com

Jonathan Hardwick
Metaswitch

Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks

Email: sesale@juniper.net

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

Tapraj Singh
Juniper Networks

Email: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Jason Walker
Comcast

Email: jason_walker2@cable.comcast.com

Nick Delregno
Verizon

Email: nick.deregno@verizon.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon

Email: joecylyn.malit@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2015

X. Xu
Huawei
C. Jacquenet
Orange
L. Fang
Microsoft

April 27, 2015

L3VPN Address Prefix Based Outbound Route Filter for BGP-4
draft-xu-bess-l3vpn-prefix-orf-02

Abstract

This document defines a new Outbound Router Filter (ORF) type for BGP, refered to as "L3VPN Address Prefix Outbound Route Filter", that can be used to perform L3VPN address-prefix-based route filtering. This ORF-type supports prefix-length- or range-based matching, wild-card-based address prefix matching, as well as the exact address prefix matching for L3VPN address families. The L3VPN Address Prefix ORF is applicable in the Virtual Subnet context.

Status of This Memo

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

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

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

This Internet-Draft will expire on October 29, 2015.

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Requirements Language 2
2. Terminology 3
3. L3VPN Address Prefix ORF Encoding 3
4. L3VPN Address Prefix ORF Matching 3
5. Acknowledgements 4
6. IANA Considerations 4
7. Security Considerations 4
8. References 4
8.1. Normative References 4
8.2. Informative References 4
Authors' Addresses 4

1. Introduction

The Outbound Route Filtering (ORF) Capability defined in [RFC5291] provides a mechanism for a BGP speaker to send to its BGP peer a set of ORFs that can be used by its peer to filter its outbound routing updates to the speaker. The Address Prefix ORF defined in [RFC5292] is used to perform address-prefix-based route filtering. However, the Address Prefix ORF is not much suitable for L3VPN [RFC4364] route filtering since there is no Route-Target (RT) field contained in the Address Prefix ORF entry.

This document builds on [RFC5292] and defines a new ORF-type for BGP, referred to as "L3VPN Address Prefix Outbound Route Filter (L3VPN Address Prefix ORF)", that can be used to perform L3VPN address prefix-based route filtering. The L3VPN Address Prefix ORF supports prefix-length- or range-based matching, wild-card-based address prefix matching, as well as the exact address prefix matching for L3VPN address families. The L3VPN Address Prefix ORF is applicable to reduce the RIB size of PE routers in the Virtual Subnet [I-D.ietf-l3vpn-virtual-subnet] context.

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. Terminology

This memo makes use of the terms defined in [RFC5292] and [RFC4364].

3. L3VPN Address Prefix ORF Encoding

The ORF-Type for the L3VPN Address Prefix ORF-Type is TBD.

A L3VPN Address Prefix ORF entry includes a Route Target field in addition to those fields which have been contained in the Address Prefix ORF [RFC5292]. That's to say, a L3VPN Address Prefix ORF entry consists of the following fields <Sequence, Action, Match, Reserved, Route-Target, Minlen, Maxlen, Length, Prefix>. Note that the Prefix field here doesn't include the Route Distinguisher (RD) part of a L3VPN address prefix. For example, in the case of a VPNv4 address prefix, only the IPv4 address prefix part of that VPNv4 address prefix is contained in that Prefix field.

A L3VPN Address Prefix ORF entry is encoded as follows: the "Action", "Match" and "Reserved" fields of the entry are encoded in the common part [RFC5291], while the remaining fields of the entry are encoded in the "type specific part" [RFC5291], as shown in Figure 1. When the Action component of an ORF entry specifies REMOVE-ALL, the entry consists of only the common part.

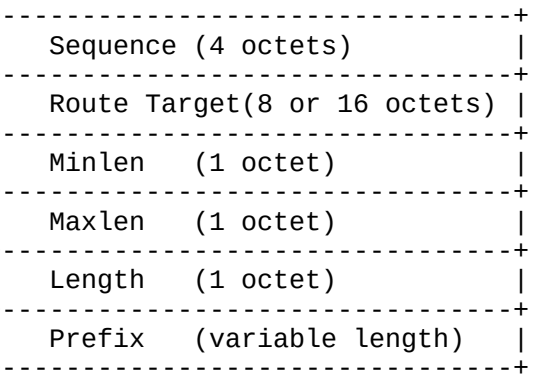


Figure 1: Type Specific Part of L3VPN Address Prefix ORF Entry Encoding

4. L3VPN Address Prefix ORF Matching

When performing route matching search on those L3VPN routes which are associated with the Route Target as specified in the received L3VPN Address Prefix ORF entries, the Address-Prefix-ORF-specific matching

rules as defined in [RFC5292] are almost preserved except that the RD SHOULD be ignored.

5. Acknowledgements

The authors would like to thank Mach Chen and Shunwan Zhuang for their comments on this document.

6. IANA Considerations

The ORF-type for the L3VPN Address Prefix ORF needs to be assigned by the IANA.

7. Security Considerations

This document does not introduce any new security considerations.

8. References

8.1. Normative References

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

[RFC5291] Chen, E. and Y. Rekhter, "Outbound Route Filtering Capability for BGP-4", RFC 5291, August 2008.

[RFC5292] Chen, E. and S. Sangli, "Address-Prefix-Based Outbound Route Filter for BGP-4", RFC 5292, August 2008.

8.2. Informative References

[I-D.ietf-l3vpn-virtual-subnet] Xu, X., Raszuk, R., Hares, S., Yongbing, F., Jacquenet, C., Boyes, T., and B. Fee, "Virtual Subnet: A L3VPN-based Subnet Extension Solution", draft-ietf-l3vpn-virtual-subnet-03 (work in progress), December 2014.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Christian Jacquenet
Orange

Email: christian.jacquenet@orange.com

Luyuan Fang
Microsoft

Email: lufang@microsoft.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 8, 2016

X. Xu
Huawei
S. Hares
Individual
Y. Fan
China Telecom
C. Jacquenet
Orange
T. Boyes
Bloomberg LP
B. Fee
Extreme Networks
August 7, 2015

RIB Reduction in Virtual Subnet
draft-xu-bess-virtual-subnet-rib-reduction-01

Abstract

Virtual Subnet is a BGP/MPLS IP VPN-based subnet extension solution which is intended for building Layer3 network virtualization overlays within and/or across data centers. This document describes a mechanism for reducing the RIB size of PE routers in the Virtual Subnet context.

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 8, 2016.

Table of Contents

1. Introduction 2
1.1. Requirements Language 3
2. Terminology 3
3. Solution Description 3
4. Acknowledgements 5
5. IANA Considerations 5
6. Security Considerations 5
7. References 5
7.1. Normative References 5
7.2. Informative References 6
Authors' Addresses 6

1. Introduction

Virtual Subnet [I-D.ietf-bess-virtual-subnet] is a BGP/MPLS IP VPN [RFC4364] -based subnet extension solution which is intended for building Layer3 network virtualization overlays within and/or across data centers. In the Virtual Subnet context, since CE host routes of a given VPN instance need to be exchanged among PE routers participating in that VPN instance, the resulting routing table size of PE routers may become a big concern, especially in large-scale data center environment where they may need to install a huge amount of host routes into their routing tables.

[I-D.ietf-bess-virtual-subnet-fib-reduction] describes a method to reduce the FIB size of PE routers without any change to the RIB and the routing table. This FIB reduction approach is applicable in the case where the control plane of PE routers still needs to maintain all host routes of the attached VPN instances for some reason (e.g., to support multicast VPN service). In the case where the control plane of PE routers doesn't need to maintain all host routes of the attached VPN instances, the RIB size of PE routers can be reduced as well which would be beneficial for CPU and memory resource saving purpose. This document proposes a very simple RIB reduction mechanism. The basic idea of this mechanism is: remote host routes

are learnt by PE routers on demand by using the L3VPN Address Prefix ORF as described in [I-D.xu-bess-l3vpn-prefix-orf].

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. Terminology

This memo makes use of the terms defined in [RFC4364].

3. Solution Description

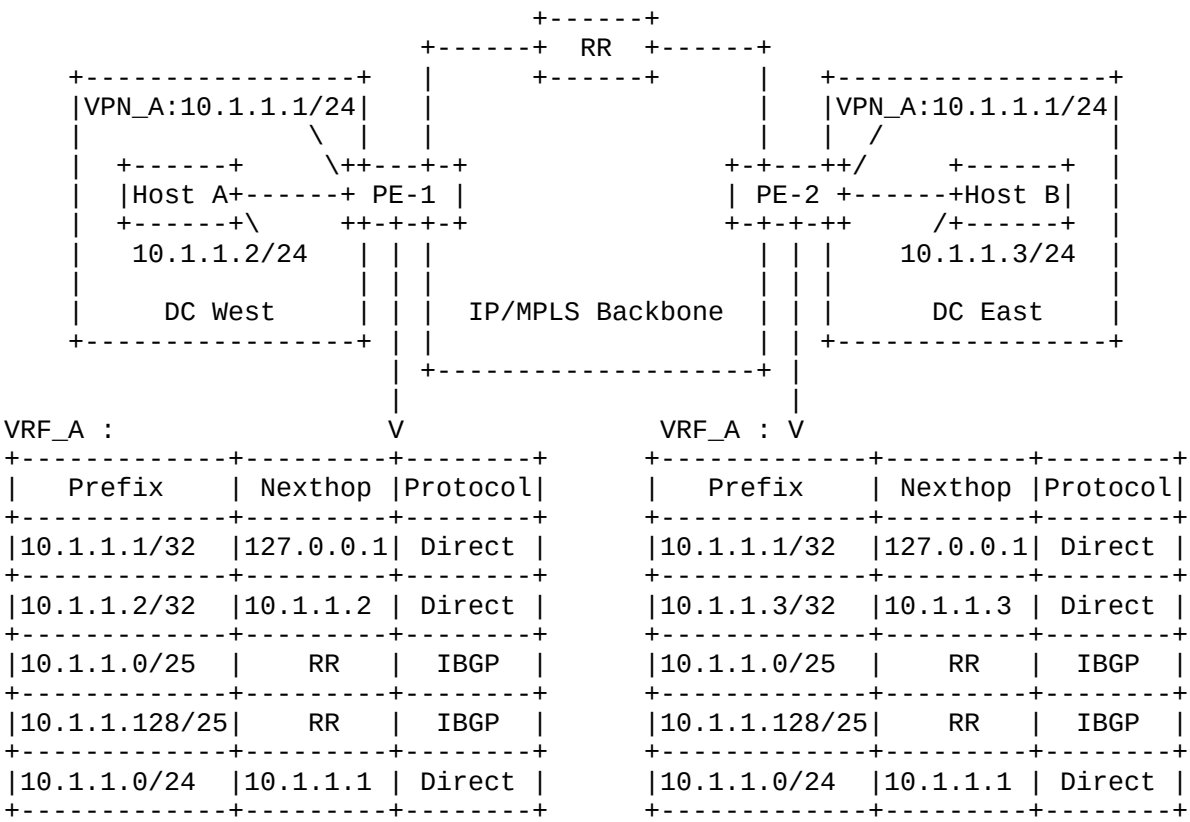


Figure 1: RIB Reduction Example

To reduce the RIB size of PE routers in the Virtual Subnet context, the L3VPN Address Prefix ORF mechanism is used to realize on-demand

route announcement. Take the VPN instance as shown in Figure 1 as an example, the RIB reduction procedures are described as follows:

1. PE routers as RR clients advertise host routes for their local CE hosts to the RR by using Rout Target (RT) ORF [RFC4364] (i.e., the RR is configured to advertise route refresh messages containing a RT-ORF entry corresponding to that VPN instance) or Route Target (RT) Constrain [RFC4684] (i.e., the RR is configured to advertise update messages containing RT membership information corresponding to that VPN instance). Those PE routers belonging to that VPN instance which don’t want to receive remote CE host routes of that VPN instance would notify the RR not to advertise any host route to them by using the L3VPN Address Prefix ORF mechanism (i.e., only requesting L3VPN routes with prefix length less than 32 (in the VPNv4 case) or 128 (in the VPNv6 case)).

2. Meanwhile, the RR is configured with static routes for more specific subnets (e.g., 10.1.1.0/25 and 10.1.1.128/25) corresponding to the extended subnet (e.g., 10.1.1.0/24) with next-hop being pointed to Null0 and then redistributes these routes to BGP. In the case where the RR is not available for transferring L3VPN traffic between PE routers for some reason (e.g., the RR is running on a server), a particular PE router other than the RR could be selected to advertise the above more specific subnet routes as long as that PE router has learnt all remote host routes belonging to that VPN instance.

3. Upon receiving a packet destined for a remote CE host from a local CE host, if there is no host route for that remote CE host in the FIB, the ingress PE router will forward the packet to the RR according to the longest-matching subnet routes learnt from the RR, which in turn forwards the packet to the relevant egress PE router according to the host route learnt from that egress PE router. As such, the RIB size of PE routers can be greatly reduced at the cost of path stretch.

4. In order to forward packets destined for that remote CE host directly to the corresponding egress PE router without any potential path stretch penalty, ingress PE routers could perform on-demand route learning of remote host routes by using one of the following options:

   A. Upon receiving an ARP request or Neighbor Solicitation (NS) message from a local CE host, if there is no CE host route for that target host in its RIB yetthe ingress PE router would request the corresponding CE host route for the target host from its RR by using the L3VPN Address Prefix ORF mechanism.

B. Upon receiving a packet whose longest-matching FIB entry is a particular more specific subnet routes (e.g., 10.1.1.0/25 and 10.1.1.128/25) learnt from the RR, a copy of this packet would be sent to the control plane while this original packet is forwarded as normal. The above copy sent to the control plane would trigger a route pull for that destination CE host. To provide robust protection against DoS attacks on the control plane, rate-limiting of the above packets sent to the control plane MUST be enabled.

5. RIB entries of remote CE host routes would expire if they have not been used for forwarding for a certain period of time. Once the expiration time for a given RIB entry is approaching, the PE router would notify its RR to remove the corresponding L3VPN Address Prefix ORF entry for that CE host route by using the L3VPN Address Prefix ORF mechanism.

4. Acknowledgements

TBD.

5. IANA Considerations

There is no requirement for any IANA action.

6. Security Considerations

This document doesn't introduce additional security risk to BGP/MPLS IP VPN, nor does it provide any additional security feature for BGP/MPLS IP VPN.

7. References

7.1. Normative References

[I-D.ietf-bess-virtual-subnet]
Xu, X., Raszuk, R., Jacquenet, C., Boyes, T., and B. Fee, "Virtual Subnet: A BGP/MPLS IP VPN-based Subnet Extension Solution", draft-ietf-bess-virtual-subnet-00 (work in progress), June 2015.

[I-D.xu-bess-l3vpn-prefix-orf]
Xu, X., Jacquenet, C., and L. Fang, "L3VPN Address Prefix Based Outbound Route Filter for BGP-4", draft-xu-bess-l3vpn-prefix-orf-02 (work in progress), April 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>.

[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>.

[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>.

7.2. Informative References

[I-D.ietf-bess-virtual-subnet-fib-reduction] Xu, X., Jacquenet, C., Boyes, T., Fee, B., and W. Henderickx, "FIB Reduction in Virtual Subnet", draft-ietf-bess-virtual-subnet-fib-reduction-01 (work in progress), July 2015.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Susan Hares
Individual

Email: shares@ndzh.com

Yongbing Fan
China Telecom

Email: fanyb@gsta.com

Christian Jacquenet
Orange

Email: christian.jacquenet@orange.com

Truman Boyes
Bloomberg LP

Email: tboyes@bloomberg.net

Brendan Fee
Extreme Networks

Email: bfee@enterasys.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

S. Zhuang
Z. Li
L. Yong
Huawei Technologies
October 19, 2015

BGP Extensions for Enhanced VPN Auto Discovery
draft-zhuang-bess-enhanced-vpn-auto-discovery-00

Abstract

All kinds of VPN technologies have been widely deployed to bear different services. As new applications develop, there proposes the requirement of auto-discovery of Layer 3 Virtual Private Network (L3VPN) and enhanced auto-discovery requirements for other VPN technologies which already have the auto-discovery mechanisms. This document identifies the possible applications and these auto-discovery requirements. Accordingly this document defines a new BGP NLRI, called the BGP-VPN-INSTANCE NLRI to satisfy the requirement of auto-discovery of BGP VPN instance and a new type of the extended community, called the Import Route Target which can be applied for auto-discovery mechanisms of multiple VPN technologies.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2016.

Table of Contents

1. Introduction 2
2. Terminologies 3
3. Requirements of VPN Auto-Discovery 3
3.1. Centralized Traffic Optimization 3
3.2. Label/Segment Allocation for VPN Instance 3
4. IRT Extended Community 4
5. BGP Extensions for L3VPN Auto-Discovery 4
5.1. BGP-VPN-INSTANCE SAFI 4
5.2. BGP-VPN-INSTANCE NLRI 5
5.2.1. VPN Membership A-D Route 6
5.3. Procedures 6
6. Contributors 7
7. IANA Considerations 7
8. Security Considerations 7
9. Acknowledgements 7
10. References 7
10.1. Normative References 7
10.2. Informative References 9
Authors' Addresses 9

1. Introduction

All kinds of VPN technologies have been widely deployed to bear different services. As new applications develop, there proposes the requirement of auto-discovery of Layer 3 Virtual Private Network (L3VPN) [RFC4364] and enhanced auto-discovery requirements for other VPN technologies which already have the auto-discovery mechanisms. This document identifies the possible applications and these auto-discovery requirements. Accordingly this document defines a new BGP NLRI, called the BGP-VPN-INSTANCE NLRI to satisfy the requirement of auto-discovery of BGP VPN instance and a new type of the extended

community, called the Import Route Target which can be applied for auto-discovery mechanisms of multiple VPN technologies.

2. Terminologies

This document uses the terminologies defined in [RFC4026]:

ERT: Export Route Target

IRT: Import Route Target

PE: Provider Edge

RD: Route Distinguisher

VRF: Virtual Routing and Forwarding

VPN A-D: VPN Auto-Discovery

3. Requirements of VPN Auto-Discovery

3.1. Centralized Traffic Optimization

As the development of central controlled application such as PCE-initiated LSP[I-D.ietf-pce-pce-initiated-lsp]and PCE-initiated P2MP LSP[I-D.palle-pce-stateful-pce-initiated-p2mp-lsp], PCE can be used to initiate setup of RSVP-TE LSP or P2MP LSP for the purpose of traffic optimization. In order to support such applications, the controller should learn the relationship of unicast VPN instances or multicast VPN instances distributed on different PEs. According to the existing auto-discovery mechanism of VPN technologies such as EVPN[RFC7432] or MVPN[RFC6514], the A-D routes are always advertised with the Export Route Target (ERT). The ingress PE can use the Import Route Target (IRT) of local MVPN/EVPN instance to match the route target advertised with the NLRI to determine the relationship of these VPN instances. But the controller which can be used as the RR of VPN routes cannot learn the relationship of VPN instances since the Import Route Target information is not advertised with these A-D routes. In order to support such applications the IRT should be carried with A-D routes.

3.2. Label/Segment Allocation for VPN Instance

[I-D.li-mpls-global-label-usecases] propose the usecases of label allocation for unicast VPN or multicast VPN instance. [I-D.li-spring-segment-path-programming] propose the usecases of segment allocation for steering traffic. In order to support such applications the PEs needs to learn the relationship of VPN instances

distributed on other PEs. For L3VPN [RFC4364] there is no auto-discovery mechanism of BGP VPN instance. In order to support such applications, auto-discovery mechanism should be introduced for L3VPN.

4. IRT Extended Community

This document defines a new type of the extended community, called as Import Route Target. This extended community is a new transitive extended community with the Sub-Type field is TBD.

The IANA registry of BGP Extended Communities clearly identifies communities of specific formats: "Two-octet AS Specific Extended Community" [RFC4360], "Four-octet AS Specific Extended Community" [RFC5668], and "IPv4 Address Specific Extended Community" [RFC4360]. Route Targets [RFC4360] extended community identify this format in the high-order (Type) octet of the Extended Community. Import Route Target extended community will reuses the same mechanism.

This document defines the following IRT Extended Communities:

Type	Sub-Type	Extended Community	Encoding
0x00	TBD	AS-2byte IRT	2-octet AS, 4-octet Value
0x01	TBD	IPv4 IRT	4-octet IPv4 Address, 2-octet Value
0x02	TBD	AS-4byte IRT	4-octet AS, 2-octet Value

Figure 1 IRT Extended Communities

The IRT Extended Community can be used for MVPN[RFC6514], L3VPN[RFC4364], EVPN[RFC7432], BGP-based VPLS[RFC4761], and BGP-AD-based VPLS[RFC6074] etc. The existing auto-discovery mechanisms of these VPN technologies always carry the ERT extended community. According to the requirements of applications, the IRT extended community SHOULD be able to be carried with different A-D routes. The local policy can be used to control the distribution of IRT information which is out of scope of this document.

5. BGP Extensions for L3VPN Auto-Discovery

5.1. BGP-VPN-INSTANCE SAFI

The BGP Multiprotocol Extensions [RFC4760] allow BGP to carry routes from multiple "address families". In this document a new Subsequent

Address Family is introduced, called "BGP-VPN-INSTANCE Sub Address Family" uses a specific BGP-VPN-INSTANCE-SAFI (TBD).

This document also defines a new BGP NLRI, called the BGP-VPN-INSTANCE NLRI to support the BGP VPN instance auto-discovery. BGP-VPN-INSTANCE MP_REACH_NLRI and MP_UNREACH_NLRI (shown in the figure 1 and figure 2) are formatted as described in [RFC4760].

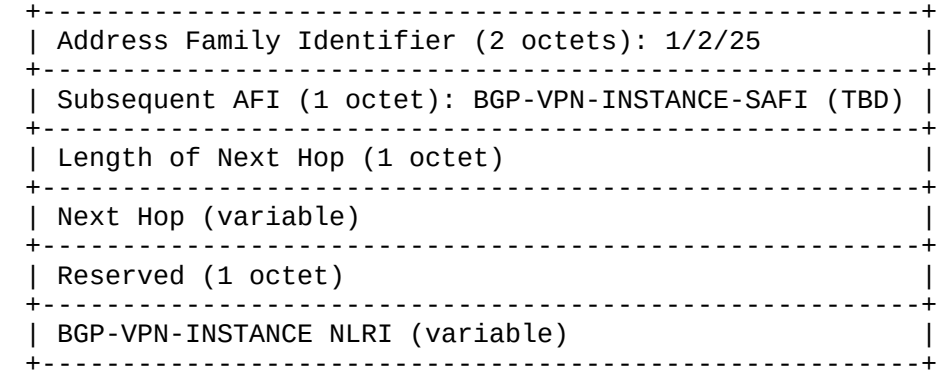


Figure 2 BGP-VPN-INSTANCE MP_REACH_NLRI

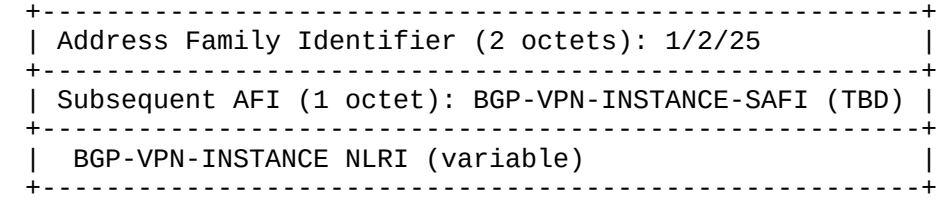


Figure 3 BGP-VPN-INSTANCE MP_UNREACH_NLRI

5.2. BGP-VPN-INSTANCE NLRI

The following is the format of the BGP-VPN-INSTANCE NLRI.

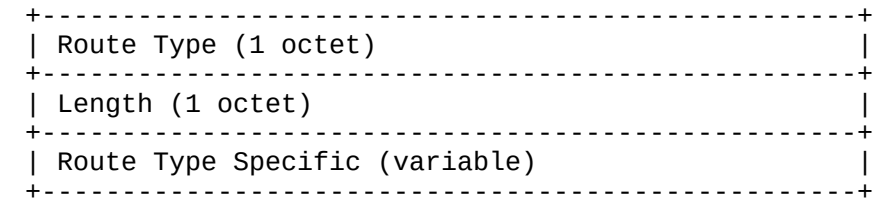


Figure 4 BGP-VPN-INSTANCE NLRI

The Route Type field defines the encoding of the rest of BGP-VPN-INSTANCE NLRI (Route Type specific BGP-VPN-INSTANCE NLRI).

The Length field indicates the length in octets of the Route Type specific field of the BGP-VPN-INSTANCE NLRI.

This document defines the following Route Types for BGP-VPN-INSTANCE routes:

-- Type 1: VPN Membership A-D Route

5.2.1. VPN Membership A-D Route

VPN Membership A-D Route is utilized for VPN Membership Auto-Discovery between PEs.

Its format is defined as following diagram:

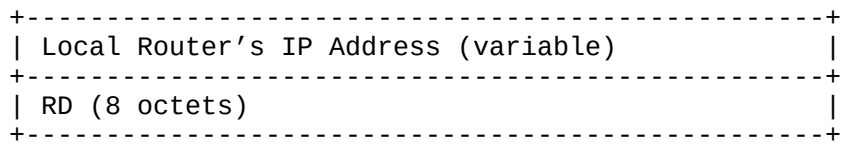


Figure 5 VPN Membership A-D Route

a) Local Router's IP Address: Advertising PE's IPv4/IPv6 address.

b) RD: RD of one VRF on advertising PE, encoded as described in [RFC4364].

5.3. Procedures

For every PE, it needs to process all its VRF configuration and generate one VPN Membership A-D Route for each VRF respectively. Local Router's IP Address field MUST filled with the Advertising Router's IP address. RD field MUST be filled with the VRF's RD value.

All ERTs of the VRF MUST be carried in BGP Update's RT Extended Community Path Attribute with the Membership A-D Route for the VRF. According to the requirement of different applications, all IRTs of the VRF SHOULD be able to be carried in BGP Update's IRT Extended Community Path Attribute with the VPN Membership A-D Route for the VRF.

If a VRF is created, then its corresponding VPN Membership A-D Route MUST be generated and advertised.

If the VRF whose VPN Membership A-D Route has been advertised is deleted, then the VPN Membership A-D Route Withdraw message MUST be generated and advertised.

If IRTs or ERTs of the VRF whose VPN Membership A-D Route has been advertised are changed, then a VPN Membership A-D Route Update with same Prefix and latest IRTs or ERTs MUST be advertised.

When the receiving PE receives VPN Membership A-D Route, VPN relationship matching MUST be checked with IRTs carried in VPN Membership A-D Route and ERTs of each Local VRF.

When the central controller receives VPN Membership A-D Route, VPN relationship matching MUST be checked with IRTs and ERTs carried in VPN Membership A-D Routes of different VPN instances.

6. Contributors

The following people have substantially contributed to the solution and to the editing of this document:.

Hui Ni
Huawei
Email: nihui@huawei.com

7. IANA Considerations

TBD.

8. Security Considerations

TBD

9. Acknowledgements

The authors would like to thank Shuanglong Chen, Eric Wu for their contributions to this work.

10. References

10.1. 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>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <http://www.rfc-editor.org/info/rfc4271>.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <http://www.rfc-editor.org/info/rfc4360>.

[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>.

[RFC4365] Rosen, E., "Applicability Statement for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4365, DOI 10.17487/RFC4365, February 2006, <http://www.rfc-editor.org/info/rfc4365>.

[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>.

[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>.

[RFC5668] Rekhter, Y., Sangli, S., and D. Tappan, "4-Octet AS Specific BGP Extended Community", RFC 5668, DOI 10.17487/RFC5668, October 2009, <http://www.rfc-editor.org/info/rfc5668>.

[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>.

[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>.

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

10.2. Informative References

[I-D.ietf-pce-pce-initiated-lsp]
Crabbe, E., Minei, I., Sivabalan, S., and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model", draft-ietf-pce-pce-initiated-lsp-04 (work in progress), April 2015.

[I-D.li-mpls-global-label-usecases]
Li, Z., Zhao, Q., Yang, T., Raszuk, R., and L. Fang, "Usecases of MPLS Global Label", draft-li-mpls-global-label-usecases-03 (work in progress), October 2015.

[I-D.li-spring-segment-path-programming]
Li, Z. and I. Milojevic, "Segment Path Programming (SPP)", draft-li-spring-segment-path-programming-00 (work in progress), October 2015.

[I-D.palle-pce-stateful-pce-initiated-p2mp-lsp]
Palle, U., Dhody, D., Tanaka, Y., Ali, Z., and V. Beeram, "PCEP Extensions for PCE-initiated Point-to-Multipoint LSP Setup in a Stateful PCE Model", draft-palle-pce-stateful-pce-initiated-p2mp-lsp-06 (work in progress), June 2015.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <http://www.rfc-editor.org/info/rfc6513>.

Authors' Addresses

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

Email: jie.dong@huawei.com

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

Email: lizhenbin@huawei.com

Lucy Yong
Huawei Technologies

Email: lucy.yong@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

S. Zhuang
W. Hao
Z. Li
Huawei Technologies

October 19, 2015

Using BGP between PE and CE in EVPN
draft-zhuang-bess-evpn-pe-ce-00

Abstract

This document identifies the possible applications which can benefit from MAC learning through the control plane between PEs and CEs. Then this document specifies protocols and procedures of using BGP as PE-CE control protocol for carrying customer MAC routing information.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 21, 2016.

Table of Contents

1. Introduction 2
2. Terminology 2
3. Applications 3
3.1. DCI Traffic Optimization 3
3.2. Inter-AS EVPN Option-A Solution 4
3.3. Fast Convergence 4
4. BGP EVPN NLRI Extensions 6
5. Exchanging C-MAC Routes 6
5.1. Originating MAC Route at the CE router 6
5.2. Receiving a MAC Route by the PE router 8
6. Contributors 8
7. IANA Considerations 8
8. Security Considerations 8
9. References 8
9.1. Normative References 9
9.2. References 9
Authors' Addresses 9

1. Introduction

[RFC7432] describes protocols and procedures for BGP MPLS based Ethernet VPNs. BGP is used for MAC learning by exchanging customer MAC routing information between PEs in the control plane instead of MAC learning between PEs in the data plane. It also states that MAC learning between PEs and CEs MAY be done in the control plane, but it does not define the detailed protocols and procedures. This document identifies the possible applications which can benefit from MAC learning through the control plane between PEs and CEs. Then this document specifies protocols and procedures of using BGP as PE-CE control protocol for carrying customer MAC routing information.

2. Terminology

This document uses terminology described in [RFC7432].

3. Applications

3.1. DCI Traffic Optimization

Figure 1 describes the Data Center Interconnect (DCI) solution when the GW and WAN PE functions are implemented in different systems.

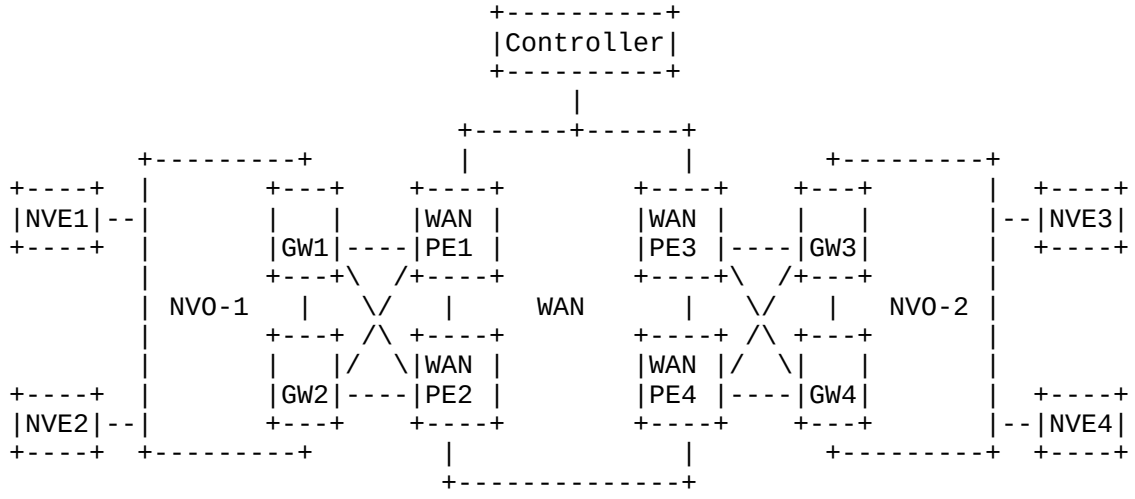


Figure 1 DCI Traffic Optimization

In the reference model depicted by Figure 1, all WAN PE routers run BGP and are connected by a Controller. For each GW, it multihoming connects to the WAN PEs, in this scenario, GW acts as an EVPN CE and WAN PE acts as an EVPN PE.

1. Requirements of outbound traffic control:

Outbound traffic control adjusts the transmission paths of outbound traffic from the WAN network to ensure that the traffic is evenly shared among PEs/links between WAN PEs and NVO networks and the bandwidth usage of each PE/link is below the specified threshold.

In outbound traffic control scenario, if the bandwidth usage of a link exceeds the specified threshold, the Controller automatically identifies which traffic needs to be scheduled and the Controller automatically calculates traffic control paths based on network topology and traffic information.

For such requirements, if the MAC routing learning between PEs and CEs or can be done through the control plane, Controller can control the multiple paths to the same destination which are receiving from

different GWs and decide which MAC route to be used for outbound traffic.

2. Requirements of Inbound traffic control:

Inbound traffic control adjusts the transmission paths of traffic bound for the WAN network to ensure that the traffic is evenly shared among PEs/links between GWs and WAN PEs and the bandwidth usage of each PE/link is below the specified threshold.

For such requirements, if the MAC routing learning between PEs and CEs or can be done through the control plane, the controller can control the path attributes of the EVPN MAC route that is advertised to the different GWs and steer the inbound traffic.

3.2. Inter-AS EVPN Option-A Solution

Currently, a typical connection mechanism between two EVPN networks can be similar to Inter-AS Option-A of [RFC4364]. In Option-A Inter-AS solution, peering ASBRs are connected by multiple sub-interfaces, each ASBR acts as a PE, and thinks that the other ASBR is a CE. For tradition L3VPN, Inter-AS Option-A has been widely deployed and MP-BGP is always adopted between ASBRs to learn IP routes. If the EVPN is introduced, there will be propose the inconsistency that IP route can be learned through the control plane while the MAC route will be learned through the forwarding plane. This will propose the challenge caused by the complex the operation and management. So in Inter-AS EVPN Option-A solution, using BGP between ASBRs, the operators can get following benefits:

1. Learning of MAC Addresses can be controlled via Peer-Based Policy between ASBRs.

2. Unified Control-Plane for MAC routing information.

3.3. Fast Convergence

The following illustrates the benefits with an example of fast convergence in the event of PE to CE network failure.

[RFC7432] defines a mechanism to efficiently and quickly signal, to remote PE nodes, the need to update their forwarding tables upon the occurrence of a failure in connectivity to an Ethernet Segment. This mechanism optimizes the withdrawal of MAC Advertisement routes, and then optimizes the network convergence time in the event of PE to CE failures. But it still cannot fully provide convergence time that is independent of the number of MAC addresses learned by the PE. There exist a situation where the network convergence time is dependent on

the local MAC learning of PE and the advertisement of them to remote PE.

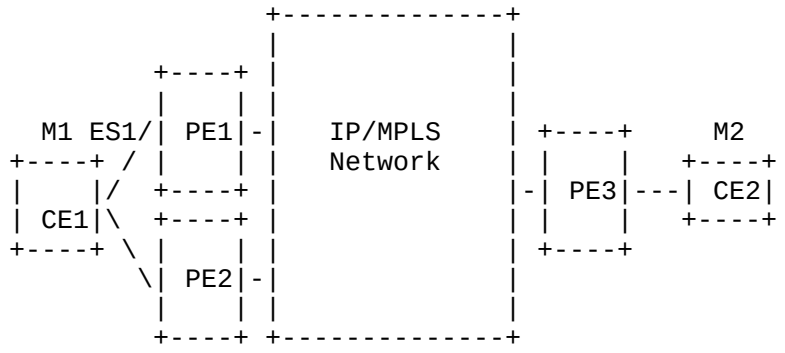


Figure 2 Multi-homed EVPN Network

To illustrate this with an example in the Figure 2, consider two PEs (PE1 and PE2) connected to a multi-homed Ethernet Segment ES1. All-Active redundancy mode is assumed. A given MAC address M1 is learned by PE1 but not PE2. On PE3, the following states may arise:

- T1- PE3 receives the Ethernet A-D routes per ESI from PE1 and PE2.
- T2- When the MAC Advertisement Route from PE1 and the Ethernet A-D routes per ESI from PE1 and PE2 are received, PE3 can forward traffic destined to M1 to both PE1 and PE2.
- T3- After T2, when the ES1 connected to PE1 fails, PE1 MUST withdraw its Ethernet A-D route per ESI, then PE3 forwards traffic destined to M1 to PE2 only.
- T4- After T3, PE1 MUST also withdraw the MAC advertisement routes (M1) that are impacted by the failure. Before PE2 learns M1 and advertises a MAC route for M1, PE3 will treat traffic to M1 as unknown unicast. If the behavior is to drop the unknown unicast based on administrative policy, the traffic to M1 on PE3 will be interrupted. Note that had PE2 also advertised a MAC route for M1 before PE1 withdraws its MAC route, then PE3 would have continued forwarding traffic destined to M1.

In the above example, once the local MAC learning of PE was done via control plane, both PE1 and PE2 will advertise a MAC Advertisement route for M1, then PE3 could continue forwarding traffic destined to M1 in the event of ES1 connected to PE1 or PE2 fails. In this case, the network convergence time is not dependent of the local MAC learning and advertisement of MAC addresses learned by the PE any more.

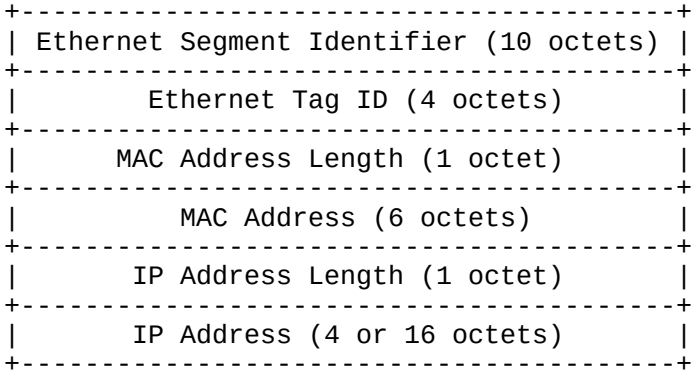
The benefit can also be achieved in case of single-active redundancy mode.

4. BGP EVPN NLRI Extensions

A new route type is defined for EVPN NLRI to advertise customer MAC route between PE and CE in EVPN:

+ 6 - Customer MAC Advertisement route

A customer MAC Advertisement route type specific EVPN NLRI consists of the following:



It should be noted that the Route Distinguisher (RD) is not used since the customer MAC routes are always exchanged in the context of unawareness of Ethernet VPN.

Another solution option is to reuse EVPN MAC Advertisement Route defined in [RFC7432] to exchange MAC route information between CE and PE. In this case RD, MPLS Label1 and MPLS Label2 fields SHOULD be set as 0. In addition, the RT for the route SHOULD also be set as 0.

5. Exchanging C-MAC Routes

This section describes the procedures of exchanging customer MAC routes between PE and CE. This document assumes that a CE and a PE exchange MAC routes over a direct BGP session.

5.1. Originating MAC Route at the CE router

When a CE receives packets in a given VLAN from interfaces, other than interfaces connected to the PE, it learns MAC addresses in the data plane. If the given VLAN is in the setting of VLANs across the Ethernet links attached to a given PE, the CE MAY advertises the MAC

addresses it learns in the data plane to the given PE, using MP-BGP and the specific MAC Route, in the control plane. The MAC Route is constructed as follows:

+ The field of the Ethernet Segment Identifier is reserved for future use.

+ The Ethernet Tag ID is set to the VLAN ID from which the MAC addresses are learned.

+ The MAC address length field is in bits and it is typically set to 48. However this specification enables specifying the MAC address as a prefix; in which case, the MAC address length field is set to the length of the prefix. This provides the ability to aggregate MAC addresses if the deployment environment supports that.

+ The MAC address is set to the value of MAC address the CE learned. The encoding of a MAC address MUST be the 6-octet MAC address specified by [802.1D-ORIG] [802.1D-REV]. If the MAC address is advertised as a prefix then the trailing bits of the prefix MUST be set to 0 to ensure that the entire prefix is encoded as 6 octets.

+ The IP Address field is optional. By default, the IP Address Length field is set to 0 and the IP Address field is omitted from the route. When a valid IP address or address prefix needs to be advertised (e.g., for ARP suppression purposes or for inter-subnet switching), it is then encoded in this route. In this case, the IP Address Length field is in bits and it is the length of the IP prefix. This provides the ability to advertise IP address prefixes when the deployment environment supports that.

+ The encoding of an IP Address MUST be either 4 octets for IPv4 or 16 octets for IPv6. When the IP Address is advertised as a prefix, then the trailing bits of the prefix MUST be set to 0 to ensure that the entire prefix is encoded as either 4 or 16 octets. The length field of Ethernet NLRI is sufficient to determine whether an IP address/prefix is encoded in this route and if so, whether the encoded IP address/prefix is IPv4 or IPv6.

+ The Next Hop field of the MP_REACH_NLRI attribute of the route MUST be set to the IPv4 or IPv6 address of the advertising CE.

It should be noted that the BGP advertisement for the MAC route does not need to carry the Route Target (RT) attributes because of its unawareness of Ethernet VPN.

5.2. Receiving a MAC Route by the PE router

When a PE receives a MAC route from a CE, it learns the MAC addresses advertised in the MAC route in the control plane and associates the MAC addresses with the Ethernet Segment from which it can reach to the advertising CE and the VLAN carried in the MAC route.

The PE SHOULD install forwarding state for the associated MAC addresses based on the Ethernet Segment and VLAN inferred from the MAC route.

In addition, the PE SHOULD advertises the MAC addresses it learns from CE in the control plane, to all the other PEs in the associated EVPN instance, using MP-BGP and the MAC Advertisement route defined in [RFC7432]. For example, the PE learns a MAC address M1 on a multi-homed Ethernet Segment (ES1) and on a VLAN 10, and the VLAN 10 is bundled to EVPN A. The PE SHOULD advertise the MAC address M1 to all the other PEs in EVPN A.

The construction of the MAC Advertisement route and procedures of handling the MAC Advertisement route on receiving it are specified in [RFC7432].

6. Contributors

The following people have substantially contributed to the solution and to the editing of this document:

Junlin Zhang
Huawei
Email: jackey.zhang@huawei.com

7. IANA Considerations

This document requires IANA to assign a new route type value for EVPN NLRI.

8. Security Considerations

There are no additional security aspects beyond those of EVPN ([RFC7432]).

9. References

9.1. Normative References

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

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

9.2. References

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

Authors' Addresses

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

Email: zhuangshunwan@huawei.com

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Email: haoweiguo@huawei.com

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

Email: lizhenbin@huawei.com