BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2017

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

October 29, 2016

YANG Data Model for MPLS-based L2VPN
draft-ietf-bess-l2vpn-yang-02.txt

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on May 2, 2017.

Copyright Notice

Table of Contents

1. Introduction 2
2. Specification of Requirements 3
3. L2VPN YANG Model 3
3.1. Overview 3
3.2. L2VPN Common 7
3.2.1. ac-templates 7
3.2.2. pw-templates 7
3.3. Point-to-Point and Multipoint service 7
3.3.1. ac list 7
3.3.2. pw list 7
3.3.3. redundancy-grp choice 7
3.3.4. endpoint container 8
3.3.5. point-to-point or multipoint service 8
3.4. Operational State 8
3.5. Yang tree 8
4. YANG Module 17
5. Security Considerations 49
6. IANA Considerations 49
7. Acknowledgments 49
8. References 49
8.1. Normative References 49
8.2. Informative References 49
Appendix A. Example Configuration 52
Appendix B. Contributors 52
Authors' Addresses 53

1. Introduction

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

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

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

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

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

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

The data model uses following constructs for configuration and management:

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

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

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

2. Specification of Requirements

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

3. L2VPN YANG Model

3.1. Overview

In this version of the document, one single container, l2vpn, is defined. Within the l2vpn container, endpoint-a, endpoint-z and a list of endpoints are defined. For the point-to-point VPWS

configuration, endpoint-a and endpoint-z are used. For the multipoint service, endpoint list is used. Each endpoint contains the definitions for attachment circuit, pseudowire and a redundancy group. The yang data model for l2vpn in this document differs significantly from the previous version which used explicit containers such as VPLS and VPWS within l2vpn, which in turn contained the definition of relevent endpoints. For example, VPWS used endpoint-a and endpoint-z while VPLS used the bridge table containing a list of endpoints.

The l2vpn container also includes definition of common building blocks for redundancy-grp templates and pseudowire-templates. The operations state object holds read-only information of objects that has either been configured or dynamically created.

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

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

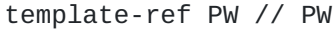
- Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC4447]
- 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]

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

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

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



```
                    template
                    attributes

 template-ref Redundancy-Group // redundancy-group
                    template
                    attributes

 l2vpn-instances // containter

         common attributes


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

         evpn-instance // reference

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

         endpoint-A // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference


         endpoint-Z // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference

         PBB-parameters // container
                           pbb specific attributes

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

 l2vpn-state // read-only container
```

Figure 1

3.2. L2VPN Common

3.2.1. ac-templates

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

3.2.2. pw-templates

The pw-templates container contains a list of pw-template. Each pw-template defines a list of common pseudowire attributes such as PW MTU, control word support etc.

3.3. Point-to-Point and Multipoint service

3.3.1. ac list

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

3.3.2. pw list

Each endpoint instance defines a list of PWs which are participating members of the given service instance. Each entry of the PW consists of one pw-template with pre-defined attributes and values, but also defines attributes that override those defined in referenced pw-template.

No restrictions are placed on type of signaling (i.e. LDP or BGP) used for a given PW. It is entirely possible to define two PWs, one signaled by LDP and other by BGP.

The VPLS specific attribute(s) are present in the definition of the PW that are member of VPLS instance only and not applicable to VPWS service.

3.3.3. redundancy-grp choice

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

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

o primary PW and backup AC

3.3.4. endpoint container

The endpoint container in general holds AC, PW or redundancy-grp references. The core aspect of endpoint container is its flexible personality based on what user decides to include in it. It is future-proofed with possible extensions that can be included in the endpoint container such as Integrated Route Bridging (IRB), PW Headend, Virtual Switch Instance, etc.

3.3.5. point-to-point or multipoint service

The point-to-point service as defined for VPWS is represented by endpoint-a and endpoint-z container that is representation of a layer 2 cross-connect of two endpoints.

The multipoint service is represented by a list of endpoints. Each endpoint container may contain a list of ACs or PWs as well as a redundancy group. The endpoint has split-horizon attribute defined and could be applicable to a list of PWs that are member of a given endpoint.

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

3.4. Operational State

The operational state of L2VPN can be queried and obtained from the read-only container defined in this document as "l2vpn-state". This container holds the runtime information of the bridge-table-instance and vpws-instance.

3.5. Yang tree

```
module: ietf-l2vpn

    +--rw l2vpn
    |  +--rw common
    |  |  +--rw pw-templates
    |  |  |  +--rw pw-template* [name]
    |  |  |     +--rw name              string
    |  |  |     +--rw mtu?              uint16
    |  |  |     +--rw cw-negotiation?   cw-negotiation-type
    |  |  |     +--rw tunnel-policy?    string
    |  |  +--rw redundancy-group-templates
    |  |     +--rw redundancy-group-template* [name]
    |  |        +--rw name               string
```

```
|  |       +--rw protection-mode?   enumeration
|  |       +--rw reroute-mode?      enumeration
|  |       +--rw dual-receive?      boolean
|  |       +--rw revert?            boolean
|  |       +--rw reroute-delay?     uint16
|  |       +--rw revert-delay?      uint16
|  +--rw l2vpn-instances
|     +--rw l2vpn-instance* [name type]
|        +--rw name                  string
|        +--rw type                  identityref
|        +--rw mtu?                  uint16
|        +--rw mac-aging-timer?      uint32
|        +--rw service-type?         l2vpn-service-type
|        +--rw discovery-type?       l2vpn-discovery-type
|        +--rw signaling-type        l2vpn-signaling-type
|        +--rw bgp-auto-discovery
|        |  +--rw route-distinguisher?   string
|        |  +--rw vpn-target* [rt-value]
|        |  |  +--rw rt-value    string
|        |  |  +--rw rt-type     bgp-rt-type
|        |  +--rw vpn-id?                string
|        +--rw bgp-signaling
|        |  +--rw site-id?      uint16
|        |  +--rw site-range?   uint16
|        +--rw pw* [name]
|        |  +--rw name              string
|        |  +--rw template?        pw-template-ref
|        |  +--rw mtu?             uint16
|        |  +--rw mac-withdraw?    boolean
|        |  +--rw cw-negotiation?  cw-negotiation-type
|        |  +--rw tunnel-policy?   string
|        |  +--rw (pw-type)?
|        |  |  +--:(ldp-or-static-pw)
|        |  |  |  +--rw peer-ip?          inet:ip-address
|        |  |  |  +--rw pw-id?            uint32
|        |  |  |  +--rw icb?              boolean
|        |  |  |  +--rw transmit-label?   mpls:mpls-label
|        |  |  |  +--rw receive-label?    mpls:mpls-label
|        |  |  +--:(bgp-pw)
|        |  |  |  +--rw remote-pe-id?     inet:ip-address
|        |  |  +--:(bgp-ad-pw)
|        |  |     +--rw remote-ve-id?     uint16
|        |  +--rw vccv-ability?     boolean
|        |  +--rw request-vlanid?   uint16
|        |  +--rw vlan-tpid?        string
|        |  +--rw ttl?              uint8
|        +--rw endpoint-a
|        |  +--rw (ac-or-pw-or-redundancy-grp)?
```

```
    |        |     +--:(ac)
    |        |     |  +--rw ac?               string
    |        |     +--:(pw)
    |        |     |  +--rw pw?               -> ../../pw/name
    |        |     +--:(redundancy-grp)
    |        |        +--rw (primary)
    |        |        |  +--:(primary-ac)
    |        |        |  |  +--rw primary-ac?       string
    |        |        |  +--:(primary-pw)
    |        |        |     +--rw primary-pw?       -> ../../pw/name
    |        |        +--rw (backup)?
    |        |        |  +--:(backup-ac)
    |        |        |  |  +--rw backup-ac?        string
    |        |        |  +--:(backup-pw)
    |        |        |     +--rw backup-pw?        -> ../../pw/name
    |        |        +--rw template?          -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
    |        |        +--rw protection-mode?   enumeration
    |        |        +--rw reroute-mode?      enumeration
    |        |        +--rw dual-receive?      boolean
    |        |        +--rw revert?            boolean
    |        |        +--rw reroute-delay?     uint16
    |        |        +--rw revert-delay?      uint16
    |        +--rw endpoint-z
    |        |  +--rw (ac-or-pw-or-redundancy-grp)?
    |        |     +--:(ac)
    |        |     |  +--rw ac?               string
    |        |     +--:(pw)
    |        |     |  +--rw pw?               -> ../../pw/name
    |        |     +--:(redundancy-grp)
    |        |        +--rw (primary)
    |        |        |  +--:(primary-ac)
    |        |        |  |  +--rw primary-ac?       string
    |        |        |  +--:(primary-pw)
    |        |        |     +--rw primary-pw?       -> ../../pw/name
    |        |        +--rw (backup)?
    |        |        |  +--:(backup-ac)
    |        |        |  |  +--rw backup-ac?        string
    |        |        |  +--:(backup-pw)
    |        |        |     +--rw backup-pw?        -> ../../pw/name
    |        |        +--rw template?          -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
    |        |        +--rw protection-mode?   enumeration
    |        |        +--rw reroute-mode?      enumeration
    |        |        +--rw dual-receive?      boolean
    |        |        +--rw revert?            boolean
    |        |        +--rw reroute-delay?     uint16
    |        |        +--rw revert-delay?      uint16
```

```
    |        +--rw pbb-parameters
    |        |  +--rw (component-type)?
```

```
    |        |     +--:(i-component)
    |        |     |  +--rw i-sid?                   i-sid-type
    |        |     |  +--rw backbone-src-mac?        yang:mac-address
    |        |     +--:(b-component)
    |        |        +--rw bind-b-component-name?   l2vpn-instance-name-ref
    |        +--rw evpn-instance?        string
    |        +--rw endpoint* [name]
    |           +--rw name                   string
    |           +--rw split-horizon-group?   string
    |           +--rw (ac-or-pw-or-redundancy-grp)?
    |              +--:(ac)
    |              |  +--rw ac* [name]
    |              |     +--rw name    string
    |              +--:(pw)
    |              |  +--rw pw* [name]
    |              |     +--rw name    -> ../../../pw/name
    |              +--:(redundancy-grp)
    |                 +--rw (primary)
    |                 |  +--:(primary-ac)
    |                 |  |  +--rw primary-ac?            string
    |                 |  +--:(primary-pw)
    |                 |     +--rw primary-pw* [name]
    |                 |        +--rw name    -> ../../../pw/name
    |                 +--rw (backup)?
    |                 |  +--:(backup-ac)
    |                 |  |  +--rw backup-ac?             string
    |                 |  +--:(backup-pw)
    |                 |     +--rw backup-pw* [name]
    |                 |        +--rw name          -> ../../../pw/name
    |                 |        +--rw precedence?   uint32
    |                 +--rw template?              -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
    |                 +--rw protection-mode?       enumeration
    |                 +--rw reroute-mode?          enumeration
    |                 +--rw dual-receive?          boolean
    |                 +--rw revert?                boolean
    |                 +--rw reroute-delay?         uint16
    |                 +--rw revert-delay?          uint16
    +--ro l2vpn-state
       +--ro l2vpn-instances
          +--ro l2vpn-instance*
             +--ro name?                string
             +--ro type?                identityref
             +--ro mtu?                 uint16
             +--ro mac-aging-timer?     uint32
             +--ro service-type?        l2vpn-service-type
             +--ro discovery-type?      l2vpn-discovery-type
             +--ro signaling-type       l2vpn-signaling-type
```

```
            +--ro bgp-auto-discovery
```

```
|  +--ro route-distinguisher?   string
|  +--ro vpn-target* [rt-value]
|  |  +--ro rt-value    string
|  |  +--ro rt-type     bgp-rt-type
|  +--ro vpn-id?                string
+--ro bgp-signaling
|  +--ro site-id?      uint16
|  +--ro site-range?   uint16
+--ro endpoint-a
|  +--ro (ac-or-pw-or-redundancy-grp)?
|     +--:(ac)
|     |  +--ro ac
|     |     +--ro name?    string
|     |     +--ro state?   operational-state-type
|     +--:(pw)
|     |  +--ro pw
|     |     +--ro name?             string
|     |     +--ro state?            operational-state-type
|     |     +--ro mtu?              uint16
|     |     +--ro mac-withdraw?     boolean
|     |     +--ro cw-negotiation?   cw-negotiation-type
|     |     +--ro vccv-ability?     boolean
|     |     +--ro tunnel-policy?    string
|     |     +--ro request-vlanid?   uint16
|     |     +--ro vlan-tpid?        string
|     |     +--ro ttl?              uint8
|     |     +--ro (pw-type)?
|     |        +--:(ldp-or-static-pw)
|     |        |  +--ro peer-ip?          inet:ip-address
|     |        |  +--ro pw-id?            uint32
|     |        |  +--ro icb?              boolean
|     |        |  +--ro transmit-label?   mpls:mpls-label
|     |        |  +--ro receive-label?    mpls:mpls-label
|     |        +--:(bgp-pw)
|     |        |  +--ro remote-pe-id?     inet:ip-address
|     |        +--:(bgp-ad-pw)
|     |           +--ro remote-ve-id?     uint16
|     +--:(redundancy-grp)
|        +--ro (primary)
|        |  +--:(primary-ac)
|        |  |  +--ro primary-ac
|        |  |     +--ro name?    string
|        |  |     +--ro state?   operational-state-type
|        |  +--:(primary-pw)
|        |     +--ro primary-pw
|        |        +--ro name?             string
|        |        +--ro state?            operational-state-type
|        |        +--ro mtu?              uint16
```

```
              |        |       +--ro mac-withdraw?     boolean
              |        |       +--ro cw-negotiation?   cw-negotiation-type
              |        |       +--ro vccv-ability?     boolean
              |        |       +--ro tunnel-policy?    string
              |        |       +--ro request-vlanid?   uint16
              |        |       +--ro vlan-tpid?        string
              |        |       +--ro ttl?              uint8
              |        |       +--ro (pw-type)?
              |        |          +--:(ldp-or-static-pw)
              |        |          |  +--ro peer-ip?          inet:ip-address
              |        |          |  +--ro pw-id?            uint32
              |        |          |  +--ro icb?              boolean
              |        |          |  +--ro transmit-label?   mpls:mpls-label
              |        |          |  +--ro receive-label?    mpls:mpls-label
              |        |          +--:(bgp-pw)
              |        |          |  +--ro remote-pe-id?     inet:ip-address
              |        |          +--:(bgp-ad-pw)
              |        |             +--ro remote-ve-id?     uint16
              |        +--ro (backup)?
              |        |  +--:(backup-ac)
              |        |  |  +--ro backup-ac
              |        |  |     +--ro name?    string
              |        |  |     +--ro state?   operational-state-type
              |        |  +--:(backup-pw)
              |        |     +--ro backup-pw
              |        |        +--ro name?             string
              |        |        +--ro state?            operational-state-type
              |        |        +--ro mtu?              uint16
              |        |        +--ro mac-withdraw?     boolean
              |        |        +--ro cw-negotiation?   cw-negotiation-type
              |        |        +--ro vccv-ability?     boolean
              |        |        +--ro tunnel-policy?    string
              |        |        +--ro request-vlanid?   uint16
              |        |        +--ro vlan-tpid?        string
              |        |        +--ro ttl?              uint8
              |        |        +--ro (pw-type)?
              |        |           +--:(ldp-or-static-pw)
              |        |           |  +--ro peer-ip?          inet:ip-address
              |        |           |  +--ro pw-id?            uint32
              |        |           |  +--ro icb?              boolean
              |        |           |  +--ro transmit-label?   mpls:mpls-label
              |        |           |  +--ro receive-label?    mpls:mpls-label
              |        |           +--:(bgp-pw)
              |        |           |  +--ro remote-pe-id?     inet:ip-address
              |        |           +--:(bgp-ad-pw)
              |        |              +--ro remote-ve-id?     uint16
              |        +--ro template?          -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
```

```
|        +--ro protection-mode?   enumeration
```

```
|        +--ro reroute-mode?      enumeration
|        +--ro dual-receive?      boolean
|        +--ro revert?            boolean
|        +--ro reroute-delay?     uint16
|        +--ro revert-delay?      uint16
+--ro endpoint-z
|  +--ro (ac-or-pw-or-redundancy-grp)?
|     +--:(ac)
|     |  +--ro ac
|     |     +--ro name?    string
|     |     +--ro state?   operational-state-type
|     +--:(pw)
|     |  +--ro pw
|     |     +--ro name?             string
|     |     +--ro state?            operational-state-type
|     |     +--ro mtu?              uint16
|     |     +--ro mac-withdraw?     boolean
|     |     +--ro cw-negotiation?   cw-negotiation-type
|     |     +--ro vccv-ability?     boolean
|     |     +--ro tunnel-policy?    string
|     |     +--ro request-vlanid?   uint16
|     |     +--ro vlan-tpid?        string
|     |     +--ro ttl?              uint8
|     |     +--ro (pw-type)?
|     |        +--:(ldp-or-static-pw)
|     |        |  +--ro peer-ip?          inet:ip-address
|     |        |  +--ro pw-id?            uint32
|     |        |  +--ro icb?              boolean
|     |        |  +--ro transmit-label?   mpls:mpls-label
|     |        |  +--ro receive-label?    mpls:mpls-label
|     |        +--:(bgp-pw)
|     |        |  +--ro remote-pe-id?     inet:ip-address
|     |        +--:(bgp-ad-pw)
|     |           +--ro remote-ve-id?     uint16
|     +--:(redundancy-grp)
|        +--ro (primary)
|        |  +--:(primary-ac)
|        |  |  +--ro primary-ac
|        |  |     +--ro name?    string
|        |  |     +--ro state?   operational-state-type
|        |  +--:(primary-pw)
|        |     +--ro primary-pw
|        |        +--ro name?             string
|        |        +--ro state?            operational-state-type
|        |        +--ro mtu?              uint16
|        |        +--ro mac-withdraw?     boolean
|        |        +--ro cw-negotiation?   cw-negotiation-type
|        |        +--ro vccv-ability?     boolean
```

```
              |        |       +--ro tunnel-policy?    string
              |        |       +--ro request-vlanid?   uint16
              |        |       +--ro vlan-tpid?        string
              |        |       +--ro ttl?              uint8
              |        |       +--ro (pw-type)?
              |        |          +--:(ldp-or-static-pw)
              |        |          |  +--ro peer-ip?          inet:ip-address
              |        |          |  +--ro pw-id?            uint32
              |        |          |  +--ro icb?              boolean
              |        |          |  +--ro transmit-label?   mpls:mpls-label
              |        |          |  +--ro receive-label?    mpls:mpls-label
              |        |          +--:(bgp-pw)
              |        |          |  +--ro remote-pe-id?     inet:ip-address
              |        |          +--:(bgp-ad-pw)
              |        |             +--ro remote-ve-id?     uint16
              |        +--ro (backup)?
              |        |  +--:(backup-ac)
              |        |  |  +--ro backup-ac
              |        |  |     +--ro name?    string
              |        |  |     +--ro state?   operational-state-type
              |        |  +--:(backup-pw)
              |        |     +--ro backup-pw
              |        |        +--ro name?             string
              |        |        +--ro state?            operational-state-type
              |        |        +--ro mtu?              uint16
              |        |        +--ro mac-withdraw?     boolean
              |        |        +--ro cw-negotiation?   cw-negotiation-type
              |        |        +--ro vccv-ability?     boolean
              |        |        +--ro tunnel-policy?    string
              |        |        +--ro request-vlanid?   uint16
              |        |        +--ro vlan-tpid?        string
              |        |        +--ro ttl?              uint8
              |        |        +--ro (pw-type)?
              |        |           +--:(ldp-or-static-pw)
              |        |           |  +--ro peer-ip?          inet:ip-address
              |        |           |  +--ro pw-id?            uint32
              |        |           |  +--ro icb?              boolean
              |        |           |  +--ro transmit-label?   mpls:mpls-label
              |        |           |  +--ro receive-label?    mpls:mpls-label
              |        |           +--:(bgp-pw)
              |        |           |  +--ro remote-pe-id?     inet:ip-address
              |        |           +--:(bgp-ad-pw)
              |        |              +--ro remote-ve-id?     uint16
              |        +--ro template?          -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
              |        +--ro protection-mode?   enumeration
              |        +--ro reroute-mode?      enumeration
              |        +--ro dual-receive?      boolean
```

```
|        +--ro revert?            boolean
```

```
|       +--ro reroute-delay?     uint16
|       +--ro revert-delay?      uint16
+--ro pbb-parameters
|  +--ro (component-type)?
|     +--:(i-component)
|     |  +--ro i-sid?                   i-sid-type
|     |  +--ro backbone-src-mac?        yang:mac-address
|     +--:(b-component)
|        +--ro bind-b-component-name?   string
|        +--ro bind-b-component-type?   identityref
+--ro evpn-instance-name?   string
+--ro endpoint*
   +--ro name?                  string
   +--ro split-horizon-group?   string
   +--ro (ac-or-pw-or-redundancy-grp)?
      +--:(ac)
      |  +--ro ac*
      |     +--ro name?    string
      |     +--ro state?   operational-state-type
      +--:(pw)
      |  +--ro pw*
      |     +--ro name?             string
      |     +--ro state?            operational-state-type
      |     +--ro mtu?              uint16
      |     +--ro mac-withdraw?     boolean
      |     +--ro cw-negotiation?   cw-negotiation-type
      |     +--ro discovery-type?   l2vpn-discovery-type
      |     +--ro signaling-type?   l2vpn-signaling-type
      |     +--ro peer-ip?          inet:ip-address
      |     +--ro pw-id?            uint32
      |     +--ro transmit-label?   mpls:mpls-label
      |     +--ro receive-label?    mpls:mpls-label
      |     +--ro tunnel-policy?    string
      +--:(redundancy-grp)
         +--ro (primary)
         |  +--:(primary-ac)
         |  |  +--ro primary-ac
         |  |     +--ro name?    string
         |  |     +--ro state?   operational-state-type
         |  +--:(primary-pw)
         |     +--ro primary-pw*
         |        +--ro name?             string
         |        +--ro state?            operational-state-type
         |        +--ro mtu?              uint16
         |        +--ro mac-withdraw?     boolean
         |        +--ro cw-negotiation?   cw-negotiation-type
         |        +--ro discovery-type?   l2vpn-discovery-type
         |        +--ro signaling-type?   l2vpn-signaling-type
```

```
                      |       +--ro peer-ip?          inet:ip-address
                      |       +--ro pw-id?            uint32
                      |       +--ro transmit-label?   mpls:mpls-label
                      |       +--ro receive-label?    mpls:mpls-label
                      |       +--ro tunnel-policy?    string
                      +--ro (backup)?
                      |  +--:(backup-ac)
                      |  |  +--ro backup-ac
                      |  |     +--ro name?    string
                      |  |     +--ro state?   operational-state-type
                      |  +--:(backup-pw)
                      |     +--ro backup-pw*
                      |        +--ro name?             string
                      |        +--ro state?            operational-state-type
                      |        +--ro mtu?              uint16
                      |        +--ro mac-withdraw?     boolean
                      |        +--ro cw-negotiation?   cw-negotiation-type
                      |        +--ro discovery-type?   l2vpn-discovery-type
                      |        +--ro signaling-type?   l2vpn-signaling-type
                      |        +--ro peer-ip?          inet:ip-address
                      |        +--ro pw-id?            uint32
                      |        +--ro transmit-label?   mpls:mpls-label
                      |        +--ro receive-label?    mpls:mpls-label
                      |        +--ro tunnel-policy?    string
                      |        +--ro precedence?       uint32
                      +--ro template?              -> /l2vpn/common/redundancy-
group-templates/redundancy-group-template/name
                      +--ro protection-mode?       enumeration
                      +--ro reroute-mode?          enumeration
                      +--ro dual-receive?          boolean
                      +--ro revert?                boolean
                      +--ro reroute-delay?         uint16
                      +--ro revert-delay?          uint16
```

Figure 2

4. YANG Module

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

```
<CODE BEGINS> file "ietf-l2vpn@2016-10-24.yang"

module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";
```

```
   import ietf-inet-types {
```

```
    prefix "inet";
  }

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

  import ietf-mpls {
    prefix "mpls";
  }

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

  revision "2016-10-24" {
    description "Fifth revision " +
                "  - Edits based on Giles's comments " +
                "    5) Remove relative leafrefs in groupings, " +
                "       and the resulting new groupings are: " +
                "       <a> bgp-auto-discovery-parameters-grp " +
                "       <b> bgp-signaling-parameters-grp " +
                "       <c> endpoint-grp " +
                "    11) Merge VPLS and VPWS into one single list " +
                "       and use augment statements to handle " +
                "       differences between VPLS and VPWS " +
                "  - Add a new grouping l2vpn-common-parameters-grp " +
                "    to make VPLS and VPWS more consistent";
    reference "";
  }

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

```
                "";
    reference "";
  }

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

  revision "2015-10-08" {
    description "Second revision " +
                "  - Added container vpls-instances " +
                "  - Rearranged groupings and typedefs to be " +
                "    reused across vpls-instance and vpws-instances";
    reference "";
  }

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

  /* identities */

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

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

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

  identity link-discovery-protocol {
    description "Base 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";
  }

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

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

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

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

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

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

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

   identity bgp-auto-discovery {
     base "l2vpn-discovery";
     description "Border Gateway Protocol (BGP) auto-discovery of " +
                 "l2vpn service";
   }

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

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

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

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

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

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

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


   /* typedefs */

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

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

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

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

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

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

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

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

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

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

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

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

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

   /* groupings */

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

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

  grouping pbb-parameters-state-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
      description "pbb-parameters";
      choice component-type {
        description "PBB component type";
        case i-component {
          leaf i-sid {
            type i-sid-type;
            description "I-SID";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component-name {
            type string;
            description "Name of the associated b-component";
          }
          leaf bind-b-component-type {
            type identityref {
              base l2vpn-instance-type;
```

```
            }
            must ". = 'vpls-instance-type'" {
              description "The associated b-component must have " +
                          "type vpls-instance-type";
            }
            description "Type of the associated b-component";
          }
        }
      }
    }
  }

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

```
   }

   grouping bgp-auto-discovery-parameters-grp {
     description "BGP parameters for auto-discovery";
     leaf route-distinguisher {
       type string;
       description "BGP RD";
     }
     list vpn-target {
       key rt-value;
       description "Route Targets";
       leaf rt-value {
         type string;
         description "Route-Target value";
       }
       leaf rt-type {
         type bgp-rt-type;
         mandatory true;
         description "Type of RT";
       }
     }
     leaf vpn-id {
       type string;
       description "VPN ID";
     }
   }

   grouping bgp-signaling-parameters-grp {
     description "BGP parameters for signaling";
     leaf site-id {
       type uint16;
       description "Site ID";
     }
     leaf site-range {
       type uint16;
       description "Site Range";
     }
   }

   grouping pw-common-parameters-grp {
     description "Pseudowire parameters common to both " +
                 "VPWS and VPLS pseudowires";
     leaf name {
       type string;
       description "pseudowire name";
     }
     leaf template {
       type pw-template-ref;
```

```
      description "pseudowire template";
    }
    leaf mtu {
      type uint16;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      default false;
      description "Enable (true) or disable (false) MAC withdraw";
    }
    leaf cw-negotiation {
      type cw-negotiation-type;
      description "cw-negotiation";
    }
    leaf tunnel-policy {
      type string;
      description "tunnel policy name";
    }
    uses pw-type-grp;
  }

  grouping pw-type-grp {
    description "pseudowire type grouping";
    choice pw-type {
      description "A choice of pseudowire type";
      case ldp-or-static-pw {
        leaf peer-ip {
          type inet:ip-address;
          description "peer IP address";
        }
        leaf pw-id {
          type uint32;
          description "pseudowire id";
        }
        leaf icb {
          type boolean;
          description "inter-chassis backup";
        }
        leaf transmit-label {
          type mpls:mpls-label;
          description "transmit lable";
        }
        leaf receive-label {
          type mpls:mpls-label;
          description "receive label";
        }
      }
```

```
      case bgp-pw {
        leaf remote-pe-id {
          type inet:ip-address;
          description "remote pe id";
        }
      }
      case bgp-ad-pw {
        leaf remote-ve-id {
          type uint16;
          description "remote ve id";
        }
      }
    }
  }

  grouping redundancy-group-properties-grp {
    description "redundancy-group-properties-grp";
    leaf protection-mode {
      type enumeration {
        enum "frr" {
          value 0;
          description "fast reroute";
        }
        enum "master-slave" {
          value 1;
          description "master-slave";
        }
        enum "independent" {
          value 2;
          description "independent";
        }
      }
      description "protection-mode";
    }
    leaf reroute-mode {
      type enumeration {
        enum "immediate" {
          value 0;
          description "immediate reroute";
        }
        enum "delayed" {
          value 1;
          description "delayed reroute";
        }
        enum "never" {
          value 2;
          description "never reroute";
        }
```

```
      }
      description "reroute-mode";
    }
    leaf dual-receive {
      type boolean;
      description
      "allow extra traffic to be carried by backup";
    }
    leaf revert {
      type boolean;
      description "allow forwarding to revert to primary " +
                  "after restoring primary";
    }
  }

  grouping endpoint-grp {
    description "A grouping that defines the structure of " +
                "an endpoint";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        description "Attachment circuit(s) as an endpoint";
      }
      case pw {
        description "Pseudowire(s) as an endpoint";
      }
      case redundancy-grp {
        description "Redundancy group as an endpoint";
        choice primary {
          mandatory true;
          description "primary options";
          case primary-ac {
            description "primary-ac";
          }
          case primary-pw {
            description "primary-pw";
          }
        }
        choice backup {
          description "backup options";
          case backup-ac {
            description "backup-ac";
          }
          case backup-pw {
            description "backup-pw";
          }
        }
```

```
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpls-pw-state-grp {
    description "vpls-pw-state-grp";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire up/down state";
    }
    leaf mtu {
      type uint16;
      description "pseudowire mtu";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (true) or disabled (false)";
    }
    leaf cw-negotiation {
      type cw-negotiation-type;
      description "cw-negotiation";
    }
    leaf discovery-type {
      type l2vpn-discovery-type;
      description "VPLS discovery type";
    }
    leaf signaling-type {
      type l2vpn-signaling-type;
      description "VPLS signaling type";
    }
    leaf peer-ip {
      type inet:ip-address;
      description "peer IP address";
    }
    leaf pw-id {
```

```
      type uint32;
      description "pseudowire id";
    }
    leaf transmit-label {
      type mpls:mpls-label;
      description "transmit lable";
    }
    leaf receive-label {
      type mpls:mpls-label;
      description "receive label";
    }
    leaf tunnel-policy {
      type string;
      description "tunnel policy name";
    }
  }

  grouping ac-state-grp {
    description "vpls-ac-state-grp";
    leaf name {
      type string;
      description "attachment circuit name";
    }
    leaf state {
      type operational-state-type;
      description "attachment circuit up/down state";
    }
  }

  grouping vpws-pw-state-grp {
    description "vpws-pw-state-grp";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire operation state up/down";
    }
    leaf mtu {
      type uint16;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (ture) or disabled (false)";
    }
    leaf cw-negotiation {
```

```
      type cw-negotiation-type;
      description "Override the control-word negotiation " +
                  "preference specified in the " +
                  "pseudowire template.";
    }
    leaf vccv-ability {
      type boolean;
      description "vccv-ability";
    }
    leaf tunnel-policy {
      type string;
      description "Used to override the tunnel policy name " +
                  "specified in the pseduowire template";
    }
    leaf request-vlanid {
      type uint16;
      description "request vlanid";
    }
    leaf vlan-tpid {
      type string;
      description "vlan tpid";
    }
    leaf ttl {
      type uint8;
      description "time-to-live";
    }
    uses pw-type-grp;
  }

  /* L2VPN YANG Model */

  container l2vpn {
    description "l2vpn";
    container common {
      description "common l2pn attributes";
      container pw-templates {
        description "pw-templates";
        list pw-template {
          key "name";
          description "pw-template";
          leaf name {
            type string;
            description "name";
          }
          leaf mtu {
            type uint16;
            description "pseudowire mtu";
          }
```

```
          leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description
              "control-word negotiation preference";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
      }
      container redundancy-group-templates {
        description "redundancy group templates";
        list redundancy-group-template {
          key "name";
          description "redundancy-group-template";
          leaf name {
            type string;
            description "name";
          }
          uses redundancy-group-properties-grp;
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to " +
                        "revert to primary";
          }
        }
      }
    }

    container l2vpn-instances {
      description "A list of L2VPN instances";
      list l2vpn-instance {
```

```
        key "name type";
        description "An L2VPN service instance";
        uses l2vpn-common-parameters-grp;
        container bgp-auto-discovery {
          when "../discovery-type = 'bgp-auto-discovery'" {
            description "Check discovery type: " +
                        "Can only configure BGP discovery if " +
                        "discovery type is BGP-AD";
          }
          description "BGP auto-discovery parameters";
          uses bgp-auto-discovery-parameters-grp;
        }
        container bgp-signaling {
          when "../signaling-type = 'bgp-signaling'" {
            description "Check signaling type: " +
                        "Can only configure BGP signaling if " +
                        "signaling type is BGP";
          }
          description "BGP signaling parameters";
          uses bgp-signaling-parameters-grp;
        }
        list pw {
          key "name";
          description "A pseudowire";
          uses pw-common-parameters-grp;
        }
      }
    }
  }

  container l2vpn-state {
    config false;
    description "l2vpn state";

    container l2vpn-instances {
      description "L2VPN instances state";
      list l2vpn-instance {
        description "An L2VPN instance's state";
        uses l2vpn-common-parameters-grp;
        container bgp-auto-discovery {
          description "BGP auto-discovery parameters";
          uses bgp-auto-discovery-parameters-grp;
        }
        container bgp-signaling {
          description "BGP signaling parameters";
          uses bgp-signaling-parameters-grp;
        }
      }
```

```
    }
  }

  /* augments */

  augment "/l2vpn/l2vpn-instances/l2vpn-instance/pw" {
    when "../type = 'vpws-l2vpn-instance-type'" {
      description "Pseudowire parameters only for VPWS pseudowires";
    }
    description "Augment for peudowire parameters for " +
                "VPWS pseudowires";
    leaf vccv-ability {
      type boolean;
      description "vccvability";
    }
    leaf request-vlanid {
      type uint16;
      description "request vlanid";
    }
    leaf vlan-tpid {
      type string;
      description "vlan tpid";
    }
    leaf ttl {
      type uint8;
      description "time-to-live";
    }
  }

  augment "/l2vpn/l2vpn-instances/l2vpn-instance" {
    when "type = 'vpws-l2vpn-instance-type'" {
      description "Endpoints specifically for a VPWS instance";
    }
    description "Augment for endpoints for a VPWS instance";
    container endpoint-a {
      description "endpoint-a";
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "An attachment circuits as the endpoint";
          leaf ac {
            type string;
            description "Name of attachment circuit.  " +
                        "This field is intended to " +
                        "reference standardized " +
                        "layer-2 definitions.";
          }
        }
```

```
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A pseudowire as the endpoint";
          leaf pw {
            type leafref {
              path "../../pw/name";
            }
            description "name of pseudowire";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-ac" {
          description "The primary attachment circuit of a " +
                      "redundancy group endpoint";
          leaf primary-ac {
            type string;
            description "Name of primary attachment circuit.  " +
                        "This field is intended to reference " +
                        "standardized layer-2 definitions.";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-pw" {
          description "The primary pseudowires of a " +
                      "redundancy group endpoint";
          leaf primary-pw {
            type leafref {
              path "../../pw/name";
            }
            description "name of pseudowire";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-ac" {
          description "The backup attachment circuit of a " +
                      "redundancy group endpoint";
          leaf backup-ac {
            type string;
            description "Name of backup attachment circuit.  " +
                        "This field is intended to reference " +
                        "standardized layer-2 definitions.";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-pw" {
          description "The backup pseudowires of a " +
                      "redundancy group endpoint";
          leaf backup-pw {
            type leafref {
```

```
              path "../../pw/name";
            }
            description "Reference an attachment circuit";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
          description "Additional conditional " +
                      "redunadancy group parameters";
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to " +
                        "revert to primary";
          }
        }
      }
    }
    container endpoint-z {
      description "endpoint-z";
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "An attachment circuits as the endpoint";
          leaf ac {
            type string;
            description "Name of attachment circuit.  " +
                        "This field is intended to " +
                        "reference standardized " +
                        "layer-2 definitions.";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A pseudowire as the endpoint";
          leaf pw {
            type leafref {
```

```
            path "../../pw/name";
          }
          description "name of pseudowire";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "primary/primary-ac" {
        description "The primary attachment circuit of a " +
                    "redundancy group endpoint";
        leaf primary-ac {
          type string;
          description "Name of primary attachment circuit.  " +
                      "This field is intended to reference " +
                      "standardized layer-2 definitions.";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "primary/primary-pw" {
        description "The primary pseudowires of a " +
                    "redundancy group endpoint";
        leaf primary-pw {
          type leafref {
            path "../../pw/name";
          }
          description "name of pseudowire";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "backup/backup-ac" {
        description "The backup attachment circuit of a " +
                    "redundancy group endpoint";
        leaf backup-ac {
          type string;
          description "Name of backup attachment circuit.  " +
                      "This field is intended to reference " +
                      "standardized layer-2 definitions.";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "backup/backup-pw" {
        description "The backup pseudowires of a " +
                    "redundancy group endpoint";
        leaf backup-pw {
          type leafref {
            path "../../pw/name";
          }
          description "Reference an attachment circuit";
        }
```

```
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
          description "Additional conditional " +
                      "redunadancy group parameters";
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to " +
                        "revert to primary";
          }
        }
      }
    }

    augment "/l2vpn/l2vpn-instances/l2vpn-instance" {
      when "type = 'vpls-l2vpn-instance-type'" {
        description "Parameters specifically for a VPLS instance";
      }
      description "Augment for parameters for a VPLS instance";
      uses pbb-parameters-grp;
      leaf evpn-instance {
        type string;
        description "Eventual reference to standard EVPN instance";
      }
      list endpoint {
        key "name";
        leaf name {
          type string;
          description "endpoint name";
        }
        leaf split-horizon-group {
          type string;
          description "Identify a split horizon group";
        }
```

```
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "A list of attachment circuits as the endpoint";
          list ac {
            key "name";
            leaf name {
              type string;
              description "Name of attachment circuit.  " +
                          "This field is intended to " +
                          "reference standardized " +
                          "layer-2 definitions.";
            }
            description "A bridge table instance's " +
                        "attachment circuit list";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A list of pseudowires as the endpoint";
          list pw {
            key "name";
            leaf name {
              type leafref {
                path "../../../pw/name";
              }
              description "name of pseudowire";
            }
            description "A VPLS instance's pseudowire list";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-ac" {
          description "The primary attachment circuit of a " +
                      "redundancy group endpoint";
          leaf primary-ac {
            type string;
            description "Name of primary attachment circuit.  " +
                        "This field is intended to reference " +
                        "standardized layer-2 definitions.";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-pw" {
          description "The primary pseudowires of a " +
                      "redundancy group endpoint";
          list primary-pw {
            key "name";
            leaf name {
```

```
            type leafref {
              path "../../../pw/name";
            }
            description "name of pseudowire";
          }
          description "A bridge table instance's pseudowire list";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "backup/backup-ac" {
        description "The backup attachment circuit of a " +
                    "redundancy group endpoint";
        leaf backup-ac {
          type string;
          description "Name of backup attachment circuit.  " +
                      "This field is intended to reference " +
                      "standardized layer-2 definitions.";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
              "backup/backup-pw" {
        description "The backup pseudowires of a " +
                    "redundancy group endpoint";
        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";
        }
      }
      augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
        description "Additional conditional " +
                    "redunadancy group parameters";
        leaf reroute-delay {
          when "../reroute-mode = 'delayed'" {
            description "Specify amount of time to " +
                        "delay reroute only when " +
                        "delayed route is configured";
          }
          type uint16;
```

```
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to revert to primary";
          }
        }
      }
      description "List of endpoints";
    }
  }

  augment "/l2vpn-state/l2vpn-instances/l2vpn-instance" {
    when "type = 'vpws-l2vpn-instance-type'" {
      description "Additional perational state specifically for " +
                  "a VPWS instance";
    }
    description "Augment for a VPWS instance's " +
                "operational state";
    container endpoint-a {
      description "endpoint-a";
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "An attachment circuit as the endpoint";
          container ac {
            description "The attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A pseudowire as the endpoint";
          container pw {
            description "The pseuduowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-ac" {
          description "The primary attachment circuit of a " +
                      "redundancy group endpoint";
          container primary-ac {
            description "The primary attachment circuit";
```

```
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-pw" {
          description "The primary pseudowires of a " +
                      "redundancy group endpoint";
          container primary-pw {
            description "The primary pseudowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-ac" {
          description "The backup attachment circuit of a " +
                      "redundancy group endpoint";
          container backup-ac {
            description "The backup attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-pw" {
          description "The backup pseudowires of a " +
                      "redundancy group endpoint";
          container backup-pw {
            description "The backup pseudowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
          description "Additional conditional " +
                      "redunadancy group parameters";
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
```

```
            type uint16;
            description "amount ot time to wait to revert to primary";
          }
        }
      }
    }
    container endpoint-z {
      description "endpoint-z";
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "An attachment circuit as the endpoint";
          container ac {
            description "The attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A pseudowire as the endpoint";
          container pw {
            description "The pseuduowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-ac" {
          description "The primary attachment circuit of a " +
                      "redundancy group endpoint";
          container primary-ac {
            description "The primary attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-pw" {
          description "The primary pseudowires of a " +
                      "redundancy group endpoint";
          container primary-pw {
            description "The primary pseudowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-ac" {
          description "The backup attachment circuit of a " +
                      "redundancy group endpoint";
          container backup-ac {
            description "The backup attachment circuit";
```

```
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-pw" {
          description "The backup pseudowires of a " +
                      "redundancy group endpoint";
          container backup-pw {
            description "The backup pseudowire";
            uses vpws-pw-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
          description "Additional conditional " +
                      "redunadancy group parameters";
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to revert to primary";
          }
        }
      }
    }

    augment "/l2vpn-state/l2vpn-instances/l2vpn-instance" {
      when "type = 'vpls-l2vpn-instance-type'" {
        description "Additional perational state specifically for " +
                    "a VPLS instance";
      }
      description "Augment for a VPLS instance's " +
                  "operational state";
      uses pbb-parameters-state-grp;
      leaf evpn-instance-name {
        type string;
```

```
      description "Name of associated an EVPN instance";
    }
    list endpoint {
      leaf name {
        type string;
        description "endpoint name";
      }
      leaf split-horizon-group {
        type string;
        description "Identify a split horizon group";
      }
      uses endpoint-grp {
        description "endpoint configuration";
        augment "ac-or-pw-or-redundancy-grp/ac" {
          description "A list of attachment circuits as the endpoint";
          list ac {
            uses ac-state-grp;
            description "The attachment circuits";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
          description "A list of pseudowires as the endpoint";
          list pw {
            uses vpls-pw-state-grp;
            description "The pseudowires";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-ac" {
          description "The primary attachment circuit of a " +
                      "redundancy group endpoint";
          container primary-ac {
            description "The primary attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "primary/primary-pw" {
          description "The primary pseudowires of a " +
                      "redundancy group endpoint";
          list primary-pw {
            uses vpls-pw-state-grp;
            description "The pseudowires";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-ac" {
          description "The backup attachment circuit of a " +
```

```
                    "redundancy group endpoint";
          container backup-ac {
            description "The backup attachment circuit";
            uses ac-state-grp;
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
                "backup/backup-pw" {
          description "The backup pseudowires of a " +
                      "redundancy group endpoint";
          list backup-pw {
            uses vpls-pw-state-grp;
            leaf precedence {
              type uint32;
              description "precedence of the pseudowire";
            }
            description "The backup pseudowires";
          }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
          description "Additional conditional " +
                      "redunadancy group parameters";
          leaf reroute-delay {
            when "../reroute-mode = 'delayed'" {
              description "Specify amount of time to " +
                          "delay reroute only when " +
                          "delayed route is configured";
            }
            type uint16;
            description "amount of time to delay reroute";
          }
          leaf revert-delay {
            when "../revert = 'true'" {
              description "Specify the amount of time to " +
                          "wait to revert to primary " +
                          "only if reversion is configured";
            }
            type uint16;
            description "amount ot time to wait to revert to primary";
          }
        }
      }
      description "List of endpoints";
    }
  }

}
```

<CODE ENDS>

Figure 3

5. Security Considerations

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

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

6. IANA Considerations

None.

7. Acknowledgments

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

8. References

8.1. Normative References

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

8.2. Informative References

[RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, DOI 10.17487/RFC3916, September 2004, <http://www.rfc-editor.org/info/rfc3916>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <http://www.rfc-editor.org/info/rfc3985>.

[RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <http://www.rfc-editor.org/info/rfc4385>.

[RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <http://www.rfc-editor.org/info/rfc4446>.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, DOI 10.17487/RFC4447, April 2006, <http://www.rfc-editor.org/info/rfc4447>.

[RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <http://www.rfc-editor.org/info/rfc4448>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <http://www.rfc-editor.org/info/rfc4664>.

[RFC4665] Augustyn, W., Ed. and Y. Serbest, Ed., "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks", RFC 4665, DOI 10.17487/RFC4665, September 2006, <http://www.rfc-editor.org/info/rfc4665>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <http://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <http://www.rfc-editor.org/info/rfc4762>.

[RFC5003] Metz, C., Martini, L., Balus, F., and J. Sugimoto, "Attachment Individual Identifier (AII) Types for Aggregation", RFC 5003, DOI 10.17487/RFC5003, September 2007, <http://www.rfc-editor.org/info/rfc5003>.

[RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, DOI 10.17487/RFC5254, October 2008, <http://www.rfc-editor.org/info/rfc5254>.

[RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <http://www.rfc-editor.org/info/rfc5659>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <http://www.rfc-editor.org/info/rfc6073>.

[RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, DOI 10.17487/RFC6074, January 2011, <http://www.rfc-editor.org/info/rfc6074>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <http://www.rfc-editor.org/info/rfc6391>.

[RFC6423] Li, H., Martini, L., He, J., and F. Huang, "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, DOI 10.17487/RFC6423, November 2011, <http://www.rfc-editor.org/info/rfc6423>.

[RFC6478] Martini, L., Swallow, G., Heron, G., and M. Bocci, "Pseudowire Status for Static Pseudowires", RFC 6478, DOI 10.17487/RFC6478, May 2012, <http://www.rfc-editor.org/info/rfc6478>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <http://www.rfc-editor.org/info/rfc6624>.

[RFC7041] Balus, F., Ed., Sajassi, A., Ed., and N. Bitar, Ed., "Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging", RFC 7041, DOI 10.17487/RFC7041, November 2013, <http://www.rfc-editor.org/info/rfc7041>.

[RFC7361] Dutta, P., Balus, F., Stokes, O., Calvignac, G., and D. Fedyk, "LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS)", RFC 7361, DOI 10.17487/RFC7361, September 2014, <http://www.rfc-editor.org/info/rfc7361>.

Appendix A. Example Configuration

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

Appendix B. Contributors

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

Reshad Rahman Cisco Systems, Inc. Email: rrahman@cisco.com Kamran Raza Cisco Systems, Inc. Email: skraza@cisco.com Tapraj Singh Cisco Systems, Inc. Email: tsingh@cisco.com Zhenbin Li Huawei Technologies Email: lizhenbin@huawei.com Zhuang Shunwan Huawei Technologies Email: Zhuangshunwan@huawei.com Wang Haibo Huawei Technologies Email:

rainsword.wang@huawei.com Sajjad Ahmed Ericsson Email: sajjad.ahmed@ericsson.com Matthew Bocci Nokia Email: matthew.bocci@nokia.com Jonathan Hardwick Metaswitch Email: jonathan.hardwick@metaswitch.com Santosh Esale Juniper Networks Email: sesale@juniper.net Kishore Tiruveedhula Juniper Networks Email: kishoret@juniper.net Nick Delregno Verizon Email: nick.deregno@verizon.com Luay Jalil Verizon Email: luay.jalil@verizon.com Maria Joecylyn Verizon Email: joecylyn.malit@verizon.com

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Ing-When Chen
Ericsson

Email: ichen@kuatrotech.com

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com