BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2016

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

March 14, 2016

YANG Data Model for MPLS-based L2VPN
draft-shah-bess-l2vpn-yang-01.txt

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

In the current version of this document, refinements to the configuration objects and Operational State objects for the same are added.

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

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

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

2. Updates in this revision

The organization of the configuration objects has been updated. The ac-templates in the common container is removed and a new redundancy-group-templates is added.

The vpls-instances container is removed and replaced with bridge-table-instances container to include the PBB, BGP parameters. This revision also introduces a reference to EVPN instance. This revision removes the definition of Attachment Circuits, "ac". Instead, the L2VPN data object model will rely on standard definitions of Attachment Circuits that IEEE and IETF are coordinating to define. Thus, this revision uses a string as a placeholder for an Attachment Circuit within the ac-or-pw-redundancy-grp within the respective endpoints list of bridge-table-instances or VPWS instances, and expect to update this field once the standard definitions of Attachment Circuits are available.

3. Specification of Requirements

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

4. L2VPN YANG Model

4.1. Overview

One single top level container, l2vpn, is defined as a parent for three different second level containers that are vpws-instances, bridge-table-instances, and common building blocks of redundancy-grp templates and pseudowire-templates. The current version of the document is extended to include refinements to configuration of vpws-instance and bridge-table-instances. The operations state object has been added to hold read-only information of objects that has either been configured or dynamically created.

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

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

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

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

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

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

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

```
 template-ref PW // PW
                    template
                    attributes

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

```
                    template
                    attributes


 bridge-table-instance name // container

         common attributes

         PBB-parameters // container
                           pbb specific attributes

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

         evpn-instance // reference

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

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

 vpws-instance name // container

         common attributes

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

         // list of PWs being used
         PW // container
                 template-ref PW
                 attribute-override
                 pw type
                    static-or-ldp
                    bgp-pw
                    bgp-ad-pw

         // ONLY 2 endpoints!!!
```

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


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

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

Figure 1

4.2. L2VPN Common

4.2.1. ac-templates

The ac-templates container is removed. The AC will eventually reference standard AC definitions defined with coordination between the IEEE and IETF, and will inherit all the attributes defined in that reference.

4.2.2. pw-templates

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

4.3. VPWS and Bridge-Table-Instance (formerly referred as VPLS)

4.3.1. ac list

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

4.3.2. pw list

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

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

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

4.3.3. redundancy-grp choice

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

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

4.3.4. endpoint container

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

The endpoint container for the VPLS service holds references to a list of ACs, a list of PWs or a redundancy group that contains a list of ACs and/or a list of PWs. This differs from the VPWS instance where an endpoint contains exactly one member; AC or PW or redundancy group and not a list.

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

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

The bridge-table-instances container contains a list of bridge-table-instance. Each entry of the bridge-table-instance represent a list of endpoints that are member of the broadcast/bridge domain. The

bridge-table-instance endpoints introduces an additional forwarding characteristics to a list of PWs and/or ACs. This split-horizon forwarding behavior is typical in bridge-table instance.

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

4.4. Operational State

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

4.5. Open items

The design team has identified several attributes that need to be included in the YANG tree. These attributes are listed here for the purpose of keeping track and are candidates for future revisions.

These attributes are :

- configuration - vccv configuration knobs
- operational state - vccv and cw-negotiation

This list is not exhaustive and expected to grow. The list will shrink as items are processed and included in the YANG tree.

4.6. Yang tree

```
module: ietf-l2vpn
   +--rw l2vpn
   |  +--rw common
   |  |  +--rw pw-templates
   |  |  |  +--rw pw-template* [name]
   |  |  |     +--rw name              string
   |  |  |     +--rw mtu?              uint32
   |  |  |     +--rw cw-negotiation?   cw-negotiation-type
   |  |  |     +--rw tunnel-policy?    string
   |  |  +--rw redundancy-group-templates
   |  |     +--rw redundancy-group-template* [name]
   |  |        +--rw name               string
   |  |        +--rw protection-mode?   enumeration
   |  |        +--rw reroute-mode?      enumeration
   |  |        +--rw reroute-delay?     uint16
   |  |        +--rw dual-receive?      boolean
   |  |        +--rw revert?            boolean
```

```
   |  |        +--rw revert-delay?      uint16
   |  +--rw bridge-table-instances
   |  |  +--rw bridge-table-instance* [name]
   |  |     +--rw name               string
   |  |     +--rw mtu?               uint32
   |  |     +--rw mac-aging-timer?   uint32
   |  |     +--rw pbb-parameters
   |  |     |  +--rw (component-type)?
   |  |     |     +--:(i-component)
   |  |     |     |  +--rw i-tag?              uint32
   |  |     |     |  +--rw backbone-src-mac?   yang:mac-address
   |  |     |     +--:(b-component)
   |  |     |        +--rw bind-b-component?   bridge-table-instance-ref
   |  |     +--rw bgp-parameters
   |  |     |  +--rw common
   |  |     |  |  +--rw route-distinguisher?   string
   |  |     |  |  +--rw vpn-targets* [rt-value]
   |  |     |  |     +--rw rt-value    string
   |  |     |  |     +--rw rt-type     bgp-rt-type
   |  |     |  +--rw discovery
   |  |     |  |  +--rw vpn-id?   string
   |  |     |  +--rw signaling
   |  |     |     +--rw site-id?      uint16
   |  |     |     +--rw site-range?   uint16
   |  |     +--rw evpn-instance?     string
   |  |     +--rw pw* [name]
   |  |     |  +--rw name              string
   |  |     |  +--rw template?         pw-template-ref
   |  |     |  +--rw mtu?              uint32
   |  |     |  +--rw mac-withdraw?     boolean
   |  |     |  +--rw cw-negotiation?   cw-negotiation-type
   |  |     |  +--rw discovery-type?   l2vpn-discovery-type
   |  |     |  +--rw signaling-type?   l2vpn-signaling-type
   |  |     |  +--rw peer-ip?          inet:ip-address
   |  |     |  +--rw pw-id?            uint32
   |  |     |  +--rw transmit-label?   uint32
   |  |     |  +--rw receive-label?    uint32
   |  |     |  +--rw tunnel-policy?    string
   |  |     +--rw endpoint* [id]
   |  |        +--rw id                      uint16
   |  |        +--rw split-horizon-group?   string
   |  |        +--rw (ac-or-pw-or-redundancy-grp)?
   |  |           +--:(ac)
   |  |           |  +--rw ac* [name]
   |  |           |     +--rw name    string
   |  |           +--:(pw)
   |  |           |  +--rw pw* [name]
   |  |           |     +--rw name    -> ../../../pw/name
```

```
   |  |           +--:(redundancy-grp)
   |  |              +--rw (primary)
   |  |              |  +--:(primary-pw)
   |  |              |  |  +--rw primary-pw* [name]
   |  |              |  |     +--rw name    -> ../../../pw/name
   |  |              |  +--:(primary-ac)
   |  |              |     +--rw primary-ac?            string
   |  |              +--rw (backup)?
   |  |              |  +--:(backup-pw)
   |  |              |  |  +--rw backup-pw* [name]
   |  |              |  |     +--rw name          -> ../../../pw/name
   |  |              |  |     +--rw precedence?   uint32
   |  |              |  +--:(backup-ac)
   |  |              |     +--rw backup-ac?            string
   |  |              +--rw template?              -> /l2vpn/common/redundancy-gr
oup-templates/redundancy-group-template/name
   |  |              +--rw protection-mode?       enumeration
   |  |              +--rw reroute-mode?          enumeration
   |  |              +--rw reroute-delay?         uint16
   |  |              +--rw dual-receive?          boolean
   |  |              +--rw revert?                boolean
   |  |              +--rw revert-delay?          uint16
   |  +--rw vpws-instances
   |     +--rw vpws-instance* [name]
   |        +--rw name               string
   |        +--rw description?       string
   |        +--rw mtu?               uint32
   |        +--rw mac-aging-timer?   uint32
   |        +--rw service-type?      l2vpn-service-type
   |        +--rw discovery-type?    l2vpn-discovery-type
   |        +--rw signaling-type     l2vpn-signaling-type
   |        +--rw bgp-parameters
   |        |  +--rw common
   |        |  |  +--rw route-distinguisher?   string
   |        |  |  +--rw vpn-targets* [rt-value]
   |        |  |     +--rw rt-value    string
   |        |  |     +--rw rt-type     bgp-rt-type
   |        |  +--rw discovery
   |        |  |  +--rw vpn-id?   string
   |        |  +--rw signaling
   |        |     +--rw site-id?      uint16
   |        |     +--rw site-range?   uint16
   |        +--rw pw* [name]
   |        |  +--rw name              string
   |        |  +--rw template?         pw-template-ref
   |        |  +--rw mtu?              uint32
   |        |  +--rw mac-withdraw?     boolean
   |        |  +--rw cw-negotiation?   cw-negotiation-type
   |        |  +--rw vccv-ability?     boolean
```

```
   |        |  +--rw tunnel-policy?    string
   |        |  +--rw request-vlanid?   uint16
   |        |  +--rw vlan-tpid?        string
   |        |  +--rw ttl?              uint8
   |        |  +--rw (pw-type)?
   |        |     +--:(ldp-or-static-pw)
   |        |     |  +--rw peer-ip?          inet:ip-address
   |        |     |  +--rw pw-id?            uint32
   |        |     |  +--rw icb?              boolean
   |        |     |  +--rw transmit-label?   uint32
   |        |     |  +--rw receive-label?    uint32
   |        |     +--:(bgp-pw)
   |        |     |  +--rw remote-pe-id?     inet:ip-address
   |        |     +--:(bgp-ad-pw)
   |        |        +--rw remote-ve-id?     uint16
   |        +--rw endpoint-a
   |        |  +--rw (ac-or-pw-or-redundancy-grp)?
   |        |     +--:(ac)
   |        |     |  +--rw ac?                string
   |        |     +--:(pw)
   |        |     |  +--rw pw?                -> ../../pw/name
   |        |     +--:(redundancy-grp)
   |        |        +--rw (primary)
   |        |        |  +--:(primary-pw)
   |        |        |  |  +--rw primary-pw?        -> ../../pw/name
   |        |        |  +--:(primary-ac)
   |        |        |     +--rw primary-ac?        string
   |        |        +--rw (backup)
   |        |        |  +--:(backup-pw)
   |        |        |  |  +--rw backup-pw?         -> ../../pw/name
   |        |        |  +--:(backup-ac)
   |        |        |     +--rw backup-ac?         string
   |        |        +--rw template?           -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |        |        +--rw protection-mode?    enumeration
   |        |        +--rw reroute-mode?       enumeration
   |        |        +--rw reroute-delay?      uint16
   |        |        +--rw dual-receive?       boolean
   |        |        +--rw revert?             boolean
   |        |        +--rw revert-delay?       uint16
   |        +--rw endpoint-z
   |           +--rw (ac-or-pw-or-redundancy-grp)?
   |              +--:(ac)
   |              |  +--rw ac?                string
   |              +--:(pw)
   |              |  +--rw pw?                -> ../../pw/name
   |              +--:(redundancy-grp)
   |                 +--rw (primary)
   |                 |  +--:(primary-pw)
```

```
   |                |  |  +--rw primary-pw?        -> ../../pw/name
   |                |  +--:(primary-ac)
   |                |     +--rw primary-ac?        string
   |                +--rw (backup)
   |                |  +--:(backup-pw)
   |                |  |  +--rw backup-pw?         -> ../../pw/name
   |                |  +--:(backup-ac)
   |                |     +--rw backup-ac?         string
   |                +--rw template?          -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |                +--rw protection-mode?   enumeration
   |                +--rw reroute-mode?      enumeration
   |                +--rw reroute-delay?     uint16
   |                +--rw dual-receive?      boolean
   |                +--rw revert?            boolean
   |                +--rw revert-delay?      uint16
   +--ro l2vpn-state
      +--ro bridge-table-instances-state
      |  +--ro bridge-table-instance-state* [name]
      |     +--ro name                  string
      |     +--ro mtu?                  uint32
      |     +--ro mac-aging-timer?      uint32
      |     +--ro pbb-parameters
      |     |  +--ro (component-type)?
      |     |     +--:(i-component)
      |     |     |  +--ro i-tag?              uint32
      |     |     |  +--ro backbone-src-mac?   yang:mac-address
      |     |     +--:(b-component)
      |     |        +--ro bind-b-component?   string
      |     +--ro bgp-parameters
      |     |  +--ro common
      |     |  |  +--ro route-distinguisher?   string
      |     |  |  +--ro vpn-targets* [rt-value]
      |     |  |     +--ro rt-value    string
      |     |  |     +--ro rt-type     bgp-rt-type
      |     |  +--ro discovery
      |     |  |  +--ro vpn-id?   string
      |     |  +--ro signaling
      |     |     +--ro site-id?      uint16
      |     |     +--ro site-range?   uint16
      |     +--ro evpn-instance-name?   string
      |     +--ro endpoint* [id]
      |        +--ro id                     uint16
      |        +--ro split-horizon-group?   string
      |        +--ro (ac-or-pw-or-redundancy-grp)?
      |           +--:(ac)
      |           |  +--ro ac* [name]
      |           |     +--ro name     string
      |           |     +--ro state?   operational-state-type
```

```
|           +--:(pw)
|           |  +--ro pw* [name]
|           |     +--ro name              string
|           |     +--ro state?            operational-state-type
|           |     +--ro mtu?              uint32
|           |     +--ro mac-withdraw?     boolean
|           |     +--ro cw-negotiation?   cw-negotiation-type
|           |     +--ro discovery-type?   l2vpn-discovery-type
|           |     +--ro signaling-type?   l2vpn-signaling-type
|           |     +--ro peer-ip?          inet:ip-address
|           |     +--ro pw-id?            uint32
|           |     +--ro transmit-label?   uint32
|           |     +--ro receive-label?    uint32
|           |     +--ro tunnel-policy?    string
|           +--:(redundancy-grp)
|              +--ro (primary)
|              |  +--:(primary-pw)
|              |  |  +--ro primary-pw* [name]
|              |  |     +--ro name              string
|              |  |     +--ro state?            operational-state-type
|              |  |     +--ro mtu?              uint32
|              |  |     +--ro mac-withdraw?     boolean
|              |  |     +--ro cw-negotiation?   cw-negotiation-type
|              |  |     +--ro discovery-type?   l2vpn-discovery-type
|              |  |     +--ro signaling-type?   l2vpn-signaling-type
|              |  |     +--ro peer-ip?          inet:ip-address
|              |  |     +--ro pw-id?            uint32
|              |  |     +--ro transmit-label?   uint32
|              |  |     +--ro receive-label?    uint32
|              |  |     +--ro tunnel-policy?    string
|              |  +--:(primary-ac)
|              |     +--ro primary-ac
|              |        +--ro name?    string
|              |        +--ro state?   operational-state-type
|              +--ro (backup)?
|              |  +--:(backup-pw)
|              |  |  +--ro backup-pw* [name]
|              |  |     +--ro name              string
|              |  |     +--ro state?            operational-state-type
|              |  |     +--ro mtu?              uint32
|              |  |     +--ro mac-withdraw?     boolean
|              |  |     +--ro cw-negotiation?   cw-negotiation-type
|              |  |     +--ro discovery-type?   l2vpn-discovery-type
|              |  |     +--ro signaling-type?   l2vpn-signaling-type
|              |  |     +--ro peer-ip?          inet:ip-address
|              |  |     +--ro pw-id?            uint32
|              |  |     +--ro transmit-label?   uint32
|              |  |     +--ro receive-label?    uint32
```

```
|              |  |     +--ro tunnel-policy?    string
|              |  |     +--ro precedence?       uint32
|              |  +--:(backup-ac)
|              |     +--ro backup-ac
|              |        +--ro name?    string
|              |        +--ro state?   operational-state-type
|              +--ro protection-mode?       enumeration
|              +--ro reroute-mode?          enumeration
|              +--ro reroute-delay?         uint16
|              +--ro dual-receive?          boolean
|              +--ro revert?                boolean
|              +--ro revert-delay?          uint16
+--ro vpws-instances-state
   +--ro vpws-instance-state* [name]
      +--ro name               string
      +--ro mtu?               uint32
      +--ro mac-aging-timer?   uint32
      +--ro service-type?      l2vpn-service-type
      +--ro discovery-type?    l2vpn-discovery-type
      +--ro signaling-type     l2vpn-signaling-type
      +--ro bgp-parameters
      |  +--ro common
      |  |  +--ro route-distinguisher?   string
      |  |  +--ro vpn-targets* [rt-value]
      |  |     +--ro rt-value    string
      |  |     +--ro rt-type     bgp-rt-type
      |  +--ro discovery
      |  |  +--ro vpn-id?   string
      |  +--ro signaling
      |     +--ro site-id?      uint16
      |     +--ro site-range?   uint16
      +--ro endpoint-a
      |  +--ro (ac-or-pw-or-redundancy-grp)?
      |     +--:(ac)
      |     |  +--ro ac
      |     |     +--ro name?    string
      |     |     +--ro state?   operational-state-type
      |     +--:(pw)
      |     |  +--ro pw
      |     |     +--ro name?             string
      |     |     +--ro state?            operational-state-type
      |     |     +--ro mtu?              uint32
      |     |     +--ro mac-withdraw?     boolean
      |     |     +--ro cw-negotiation?   cw-negotiation-type
      |     |     +--ro vccv-ability?     boolean
      |     |     +--ro tunnel-policy?    string
      |     |     +--ro request-vlanid?   uint16
      |     |     +--ro vlan-tpid?        string
```

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

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

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

```
                    |  |        |  +--ro icb?              boolean
                    |  |        |  +--ro transmit-label?   uint32
                    |  |        |  +--ro receive-label?    uint32
                    |  |        +--:(bgp-pw)
                    |  |        |  +--ro remote-pe-id?     inet:ip-address
                    |  |        +--:(bgp-ad-pw)
                    |  |           +--ro remote-ve-id?     uint16
                    |  +--:(backup-ac)
                    |     +--ro backup-ac-name?    string
                    +--ro protection-mode?   enumeration
                    +--ro reroute-mode?      enumeration
                    +--ro reroute-delay?     uint16
                    +--ro dual-receive?      boolean
                    +--ro revert?            boolean
                    +--ro revert-delay?      uint16
```

Figure 2

5. YANG Module

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

```
<CODE BEGINS> file "ietf-l2vpn@2016-03-07.yang"
module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";

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

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

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

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

```
               "  - Added state branch";
    reference "";
  }

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

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

  /* identities */

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

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

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

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

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

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

```
  /* typedefs */

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

  typedef l2vpn-discovery-type {
    type enumeration {
      enum manual {
        description "Manual configuration";
      }
      enum bgp-ad {
        description "Border Gateway Protocol (BGP) auto-discovery";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP)";
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN discovery type";
  }

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

```
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN signaling type";
  }

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

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

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

  typedef pbb-component-type {
    type enumeration {
      enum "b-component" {
        description "Identifies as a b-component";
      }
```

```
      enum "i-component" {
        description "Identifies as an i-component";
      }
    }
    description "This type is used to identify " +
                "the type of PBB component";
  }

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

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

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

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

  /* groupings */

  grouping pbb-parameters-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
```

```
      description "pbb-parameters";
      choice component-type {
        description "PBB component type";
        case i-component {
          leaf i-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type bridge-table-instance-ref;
            description "Reference to the associated b-component";
          }
        }
      }
    }
  }

  grouping pbb-parameters-state-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
      description "pbb-parameters";
      choice component-type {
        description "PBB component type";
        case i-component {
          leaf i-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type string;
            description "Name of the associated b-component";
          }
        }
      }
    }
  }
```

```
  grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
      description "Parameters for BGP";
      container common {
        when "../../discovery-type = 'bgp-ad'" {
          description "Check discovery type: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "Common BGP parameters";
        leaf route-distinguisher {
          type string;
          description "BGP RD";
        }
        list vpn-targets {
          key rt-value;
          description "Route Targets";
          leaf rt-value {
            type string;
            description "Route-Target value";
          }
          leaf rt-type {
            type bgp-rt-type;
            mandatory true;
            description "Type of RT";
          }
        }
      }
      container discovery {
        when "../../discovery-type = 'bgp-ad'" {
          description "BGP parameters for discovery: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "BGP parameters for discovery";
        leaf vpn-id {
          type string;
          description "VPN ID";
        }
      }
      container signaling {
        when "../../signaling-type = 'bgp'" {
          description "Check signaling type: " +
                      "Can only configure BGP signaling if " +
                      "signaling type is BGP";
        }
        description "BGP parameters for signaling";
```

```
        leaf site-id {
          type uint16;
          description "Site ID";
        }
        leaf site-range {
          type uint16;
          description "Site Range";
        }
      }
    }
  }

  grouping pw-type-grp {
    description "pseudowire type grouping";
    choice pw-type {
      description "A choice of pseudowire type";
      case ldp-or-static-pw {
        leaf peer-ip {
          type inet:ip-address;
          description "peer IP address";
        }
        leaf pw-id {
          type uint32;
          description "pseudowire id";
        }
        leaf icb {
          type boolean;
          description "inter-chassis backup";
        }
        leaf transmit-label {
          type uint32;
          description "transmit lable";
        }
        leaf receive-label {
          type uint32;
          description "receive label";
        }
      }
      case bgp-pw {
        leaf remote-pe-id {
          type inet:ip-address;
          description "remote pe id";
        }
      }
      case bgp-ad-pw {
        leaf remote-ve-id {
          type uint16;
          description "remote ve id";
```

```
        }
      }
    }
  }

  grouping bridge-table-instance-pw-list-grp {
    description "bridge-table-instance-pw-list-grp";
    list pw {
      key "name";
      leaf name {
        type leafref {
          path "../../../pw/name";
        }
        description "name of pseudowire";
      }
      description "A bridge table instance's pseudowire list";
    }
  }

  grouping bridge-table-instance-ac-list-grp {
    description "bridge-table-instance-ac-list-grp";
    list ac {
      key "name";
      leaf name {
        type string;
        description "Name of attachment circuit.  This field " +
                    "is intended to reference standardized " +
                    "layer-2 definitions.";
      }
      description "A bridge table instance's " +
                  "attachment circuit list";
    }
  }

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

```
          description "independent";
        }
      }
      description "protection-mode";
    }
    leaf reroute-mode {
      type enumeration {
        enum "immediate" {
          value 0;
          description "immediate reroute";
        }
        enum "delayed" {
          value 1;
          description "delayed reroute";
        }
        enum "never" {
          value 2;
          description "never reroute";
        }
      }
      description "reroute-mode";
    }
    leaf reroute-delay {
      when "../reroute-mode = 'delayed'" {
        description "Specify amount of time to delay reroute " +
                    "only when delayed route is configured";
      }
      type uint16;
      description "amount of time to delay reroute";
    }
    leaf dual-receive {
      type boolean;
      description
      "allow extra traffic to be carried by backup";
    }
    leaf revert {
      type boolean;
      description "allow forwarding to revert to primary " +
                  "after restoring primary";
      /* This is called "revertive" during the discussion. */
    }
    leaf revert-delay {
      when "../revert = 'true'" {
        description "Specify the amount of time to wait to revert " +
                    "to primary only if reversion is configured";
      }
      type uint16;
      description "amount ot time to wait to revert to primary";
```

```
      /* This is called "wtr" during discussion. */
    }
  }

  grouping bridge-table-instance-endpoint-grp {
    description "A bridge table instance's endpoint";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        uses bridge-table-instance-ac-list-grp;
        description "reference to attachment circuits";
      }
      case pw {
        uses bridge-table-instance-pw-list-grp;
        description "reference to pseudowires";
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw {
            description "primary-pw";
            list primary-pw {
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference a pseudowire";
              }
              description "A list of primary pseudowires";
            }
          }
          case primary-ac {
            description "primary-ac";
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          description "backup options";
          case backup-pw {
            list backup-pw {
```

```
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference an attachment circuit";
              }
              leaf precedence {
                type uint32;
                description "precedence of the pseudowire";
              }
              description "A list of backup pseudowires";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
            description "backup-ac";
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        leaf ac {
          type string;
          description "Name of attachment circuit.  This " +
                      "field is intended to reference " +
```

```
                    "standardized layer-2 definitions.";
        }
      }
      case pw {
        leaf pw {
          type leafref {
            path "../../pw/name";
          }
          description "reference to a pseudowire";
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            leaf primary-pw {
              type leafref {
                path "../../pw/name";
              }
              description "primary pseudowire";
            }
          }
          case primary-ac {
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            leaf backup-pw {
              type leafref {
                path "../../pw/name";
              }
              description "backup pseudowire";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
```

```
                          "standardized layer-2 definitions.";
            }
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-state-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        container ac {
          description "ac";
          uses ac-state-grp;
        }
      }
      case pw {
        container pw {
          description "pw";
          uses vpws-pw-state-grp;
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            container primary-pw {
              description "primary pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case primary-ac {
            leaf primary-ac-name {
              type string;
```

```
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            container backup-pw {
              description "backup pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case backup-ac {
            leaf backup-ac-name {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpls-pw-state-grp {
    description "vpls-pw-state-grp";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire up/down state";
    }
    leaf mtu {
      type uint32;
      description "pseudowire mtu";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (true) or disabled (false)";
    }
    leaf cw-negotiation {
```

```
      type cw-negotiation-type;
      description "cw-negotiation";
    }
    leaf discovery-type {
      type l2vpn-discovery-type;
      description "VPLS discovery type";
    }
    leaf signaling-type {
      type l2vpn-signaling-type;
      description "VPLS signaling type";
    }
    leaf peer-ip {
      type inet:ip-address;
      description "peer IP address";
    }
    leaf pw-id {
      type uint32;
      description "pseudowire id";
    }
    leaf transmit-label {
      type uint32;
      description "transmit lable";
    }
    leaf receive-label {
      type uint32;
      description "receive label";
    }
    leaf tunnel-policy {
      type string;
      description "tunnel policy name";
    }
  }

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

  grouping vpws-pw-state-grp {
    description "vpws-pw-state-grp";
    leaf name {
```

```
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire operation state up/down";
    }
    leaf mtu {
      type uint32;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (ture) or disabled (false)";
    }
    leaf cw-negotiation {
      type cw-negotiation-type;
      description "Override the control-word negotiation " +
                  "preference specified in the " +
                  "pseudowire template.";
    }
    leaf vccv-ability {
      type boolean;
      description "vccv-ability";
    }
    leaf tunnel-policy {
      type string;
      description "Used to override the tunnel policy name " +
                  "specified in the pseduowire template";
    }
    leaf request-vlanid {
      type uint16;
      description "request vlanid";
    }
    leaf vlan-tpid {
      type string;
      description "vlan tpid";
    }
    leaf ttl {
      type uint8;
      description "time-to-live";
    }
    uses pw-type-grp;
  }

  /* L2VPN YANG Model */

  container l2vpn {
```

```
    description "l2vpn";
    container common {
      description "common l2pn attributes";
      container pw-templates {
        description "pw-templates";
        list pw-template {
          key "name";
          description "pw-template";
          leaf name {
            type string;
            description "name";
          }
          leaf mtu {
            type uint32;
            description "pseudowire mtu";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description
              "control-word negotiation preference";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
      }
      container redundancy-group-templates {
        description "redundancy group templates";
        list redundancy-group-template {
          key "name";
          description "redundancy-group-template";
          leaf name {
            type string;
            description "name";
          }
          uses redundancy-group-properties-grp;
        }
      }
    }
    container bridge-table-instances {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances";
      list bridge-table-instance {
        key "name";
        description "A bridge table instance";
        leaf name {
```

```
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance {
          type string;
          description "Eventual reference to standard EVPN instance";
        }
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            description "cw-negotiation";
          }
          leaf discovery-type {
            type l2vpn-discovery-type;
            description "VPLS discovery type";
          }
          leaf signaling-type {
            type l2vpn-signaling-type;
            description "VPLS signaling type";
```

```
          }
          leaf peer-ip {
            type inet:ip-address;
            description "peer IP address";
          }
          leaf pw-id {
            type uint32;
            description "pseudowire id";
          }
          leaf transmit-label {
            type uint32;
            description "transmit lable";
          }
          leaf receive-label {
            type uint32;
            description "receive label";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          uses bridge-table-instance-endpoint-grp;
          description "List of endpoints";
        }
      }
    }
    container vpws-instances {
      description "vpws-instances";
      list vpws-instance {
        key "name";
        description "A VPWS instance";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf description {
          type string;
```

```
          description "Description of the VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
```

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

  container l2vpn-state {
    config false;
    description "l2vpn state";
    container bridge-table-instances-state {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances-state";
      list bridge-table-instance-state {
        key "name";
```

```
        description "A bridge table instance's state data";
        leaf name {
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-state-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance-name {
          type string;
          description "Name of associated an EVPN instance";
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          choice ac-or-pw-or-redundancy-grp {
            description "A choice ofattachment circuit or " +
                        "pseudowire or redundancy group";
            case ac {
              list ac {
                key "name";
                uses ac-state-grp;
                description "A list of attachment circuits";
              }
              description "attachment circuit endpoint state";
            }
            case pw {
              list pw {
                key "name";
                uses vpls-pw-state-grp;
                description "A list of pseudowires";
              }
              description "pseudowire endpoint state";
            }
```

```
            case redundancy-grp {
              choice primary {
                mandatory true;
                description "primary options";
                case primary-pw {
                  description "primary-pw";
                  list primary-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    description "A list of primary pseudowires";
                  }
                }
                case primary-ac {
                  description "primary-ac";
                  container primary-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              choice backup {
                description "backup options";
                case backup-pw {
                  list backup-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    leaf precedence {
                      type uint32;
                      description "precedence of the pseudowire";
                    }
                    description "A list of backup pseudowires";
                  }
                }
                case backup-ac {
                  description "backup-ac";
                  container backup-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              uses redundancy-group-properties-grp;
            }
          }
          description "List of endpoints";
        }
      }
    }
```

```
    container vpws-instances-state {
      description "vpws-instances-state";
      list vpws-instance-state {
        key "name";
        description "A VPWS instance's state data";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        container endpoint-a {
          description "endpoint-a";
          uses vpws-endpoint-state-grp;
        }
        container endpoint-z {
          description "endpoint-z";
          uses vpws-endpoint-state-grp;
        }
      }
    }
  }
}
```

<CODE ENDS>

Figure 3

6. Security Considerations

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

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

7. IANA Considerations

None.

8. Acknowledgments

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

9. References

9.1. Normative References

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

9.2. Informative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Reshad Rahman
Cisco Systems, Inc.

Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.

Email: skraza@cisco.com

Zhenbin Li
Huawei Technologies

Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies

Email: Zhuangshunwan@huawei.com

Wang Haibo
Huawei Technologies

Email: rainsword.wang@huawei.com

Ing-When Chen
Ericsson

Email: ichen@kuatrotech.com

Sajjad Ahmed
Ericsson

Email: sajjad.ahmed@ericsson.com

Mathew Bocci
Alcatel-Lucent

Email: mathew.bocci@alcatel-lucent.com

Jonathan Hardwick
Metaswitch

Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks

Email: sesale@juniper.net

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

Tapraj Singh
Juniper Networks

Email: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Jason Walker
Comcast

Email: jason_walker2@cable.comcast.com

Nick Delregno
Verizon

Email: nick.deregno@verizon.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon

Email: joecylyn.malit@verizon.com