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 September 14, 2017.
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 are expected to cover the actions and notifications aspects of the model.

The L2VPN data object model uses the instance centric approach. Within an L2VPN instance; a set of common parameters, a list of PWs and a list of endpoints are defined. A special constraint is added for the VPWS configuration such that only two endpoints are allowed in the list of endpoints. This deviates from the previous versions where endpoint-a and endpoint-z were defined separately from the endpoint list.
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, for configuration, one single container, l2vpn, is defined. Within the l2vpn container, common parameters and a list of endpoints are defined. For the point-to-point VPWS configuration, endpoint list is used with the constraint that limits the number of endpoints to be two. For the multipoint service, endpoint list is used. Each endpoint contains the common definition that is either an attachment circuit, a pseudowire or a redundancy group. The YANG data model for l2vpn in this document is greatly simplified by by removing separate definition of endpoint-a and endpoint-z that was specific for VPWS service. The same endpoint list is used by both the VPLS and VPWS service with the exception that VPWS uses only two entries.

   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:

   o Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]

   o Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]

   o IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- 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]
o Pseudowire status for static pseudowire [RFC6478]

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

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

The following is the high level view of the L2VPN data model.
template-ref PW // PW
  template
  attributes

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

l2vpn-instances // container
  common attributes

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

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

PBB-parameters // container
  pbb specific attributes

VPWS-constraints // rule to limit number of endpoints to two

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

l2vpn-state // read-only container

Figure 1

3.2. Open issues and next steps

There are a number of additional definitions that are in considerations. These are VPLS IRB, PW headend, how evpn instance is referenced within the scope of l2vpn instance and what parameters of evpn are defined in l2vpn data model. Some of these are local and remote VPWS service IDs, FXC, Designated Forwarder priorities, etc.
The contributors of this document intend to close on these definitions during the ongoing design team meeting as well as face-to-face meetings at the IETF.

3.3. L2VPN Common

3.3.1. 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.

3.3.2. redundancy-group-templates

The redundancy-group-templates contains a list of templates. Each template defines common attributes related to redundancy such as protection mode, reversion parameters, etc.

3.4. L2VPN instance

A list of L2VPN instance is defined where each entry represent a point to point or multipoint service. Within a service instance, a set of common attributes are defined, followed by a list of PWs and a list of endpoints.

3.4.1. common attributes

The common attributes apply to entire L2VPN instance. These attributes typically include attributes such as mac-aging-timer, BGP related parameters (if using BGP signaling), discovery-type, etc.

3.4.2. PW list

The PW list is the number of PWs that are being used for a given L2VPN instance. Each PW entry refers to PW template to inherit common attributes for the PW. The one or more attributes from the template can be overriden. It further extends definitions of more PW specific attributes such as use of control word, mac withdraw, what type of signaling (i.e. LDP or BGP), setting of the TTL, etc.

3.4.3. List of endpoints

The list of endpoints define the characteristics of the L2VPN service. In the case of VPWS, the list is limited to two entries while for VPLS, there could be many.

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

The endpoint entry also defines the split-horizon attribute which defines the frame forwarding restrictions between the endpoints belonging to same split-horizon group. This construct permits multiple instances of split horizon groups with its own endpoint members. The frame forwarding restrictions does not apply between endpoints that belong to two different split horizon groups.

3.4.3.1. ac

Attachment Circuit (AC) resides within endpoint entry either as an independent entity or as a member of the redundancy group. AC is not defined in this document but references the definitions being specified by other working groups and standard bodies.

3.4.3.2. pw

The Pseudo-wire resides within endpoint entry either as an independent entity or as a member of the redundancy group. The PW refers to one of the entry in the list of PWs defined with the L2VPN instance.

3.4.3.3. 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

The redundancy group also defines attributes of the type of redundancy, such as protection mode, reroute mode, reversion related parameters, etc.
3.4.4. point-to-point or multipoint service

The point-to-point service as defined for VPWS is represented by a list of endpoints and is limited to two entries by the VPWS constrain rules.

The multipoint service as defined for VPLS is represented by a list of endpoints.

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

3.5. 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.

3.6. Yang tree

module: ietf-l2vpn
  +--rw l2vpn
    |  +--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?        12vpn-service-type
    |     |     |     +--rw discovery-type?      12vpn-discovery-type
    |     |     |     +--rw signaling-type       12vpn-signaling-type
bgp-auto-discovery
  route-distinguisher? rt-types:route-distinguisher
  vpn-id? string
  vpn-target* [route-target]
    route-target rt-types:route-target
    route-target-type rt-types:route-target-type
bgp-signaling
  site-id? uint16
  site-range? uint16
pw* [name]
  name string
  template? pw-template-ref
  mtu? uint16
  mac-withdraw? boolean
  cw-negotiation? cw-negotiation-type
  tunnel-policy? string
  (pw-type)?
    (ldp-or-static-pw)
      peer-ip? inet:ip-address
      pw-id? uint32
      icle boolean
      transmit-label? rt-types:mpls-label
      receive-label? rt-types:mpls-label
    (bgp-pw)
      remote-pe-id? inet:ip-address
    (bgp-ad-pw)
      remote-ve-id? uint16
  vccv-ability? boolean
  request-vlanid? uint16
  vlan-tpid? string
  ttl? uint8
endpoint* [name]
  name string
  (ac-or-pw-or-redundancy-grp)?
    (ac)
      ac* [name]
      name string
    (pw)
      pw* [name]
      name -> ../../../pw/name
    (redundancy-grp)
      primary
        primary-ac?
        primary-pw
          primary-pw* [name]
          name -> ../../../pw/name
      backup
        name -> ../../../pw/name
<null>
Internet-Draft          YANG Data Model for L2VPN             March 2017

++-:(pw)
   +=-:(pw)
     +=-:(ldp-or-static-pw)
     +=-:(bgp-pw)
     +=-:(bgp-ad-pw)
     +=-:(primary-ac)
     +=-:(primary-pw)
     +=-:(redundancy-grp)

++-:(primary-ac)
   +=-:(primary-pw)
     +=-:(ldp-or-static-pw)
     +=-:(bgp-pw)
     +=-:(bgp-ad-pw)

++-:(redundancy-grp)
**YANG Data Model for L2VPN**

```
++-ro vccv-ability?         boolean  
++-ro request-vlanid?       uint16  
++-ro vlan-tpid?            string  
++-ro ttl?                  uint8   
+++ro (backup)?             
+++ro (backup-ac)           
  +++ro name?                string  
  +++ro state?               operational-state-type  
+++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 tunnel-policy?       string  
  +++ro (pw-type)?           
    |  +++ro (ldp-or-static-pw)
    |    |  +++ro peer-ip?          inet:ip-address  
    |    |  +++ro pw-id?            uint32  
    |    |  +++ro ich?              boolean  
    |    |  +++ro transmit-label?   rt-types:mpls-label  
    |    |  +++ro receive-label?    rt-types:mpls-label  
    |    |  +++ro (bgp-pw)
    |    |    |  +++ro remote-pw-id?     inet:ip-address  
    |    |    |  +++ro remote-ve-id?     uint16  
    |    |    |  +++ro vccv-ability?    boolean  
    |    |    |  +++ro request-vlanid?   uint16  
    |    |    |  +++ro vlan-tpid?       string  
    |    |    |  +++ro ttl?              uint8   
    |    |    |  +++ro precedence?       uint32  
    |    |    |  +++ro (component-type)? 
    |    |    |    |  +++ro (i-component)
    |    |    |    |    |  +++ro i-sid?             i-sid-type  
    |    |    |    |    |  +++ro backbone-src-mac?  yang:mac-address  
    |    |    |    |    |  +++ro (b-component)
    |    |    |    |    |    |  +++ro bind-b-component-name? string
```
4. YANG Module

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

<CODE BEGINS> file "ietf-l2vpn@2017-03-06.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";
    }

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

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

    revision "2017-03-06" {
        description "Sixth revision " +
            " - Removed the 'common' container and move pw-templates " +
            " and redundancy-group-templates up a level " +
            " - Consolidated the endpoint configuration such that " +
            " all L2VPN instances has a list of endpoint. For " +
            " certain types of L2VPN instances such as VPWS where " +
            " each L2VPN instance is limited to at most two " +
            " endpoint, additional augment statements were included " +
            " to add necessary constraints " +
            " - Removed discovery-type and signaling-type operational " +
            " state from VPLS pseudowires, as these two parameters " +
        }

        

are configured as L2VPN parameters rather than "+
    pseudowire parameters" +
- Renamed l2vpn-instances to l2vpn-instances-state "+
in the operational state branch" +
- Removed BGP parameter groupings and reused "+
  ietf-routing-types.yang module instead" +
"; reference ";
}

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

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

revision "2016-03-07" {

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

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

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

/* identities */

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

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

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

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

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

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

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

identity udld {
    base "link-discovery-protocol";
    description "This identity represents 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 represents Frame-Relay service";
}

identity TDM {
    base "l2vpn-service";
    description "This identity represents Time Division " +
    "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 {
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 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/pw-templates/pw-template/name";
    }
    description "pw-template-ref";
}

typedef redundancy-group-template-ref {
    type leafref {
        path "/l2vpn/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 12vpn-instance-name-ref;
                    must "/12vpn" +
                        "/12vpn-instances/12vpn-instance[name=current()]" +
                        "/type = 'vpls-instance-type'" {
                        description "A b-component must be an L2VPN instance " +
                            "of type vpls-instance-type";
                    }
                    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-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-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 rt-types:mpls-label;
                description "transmit label";
            }
            leaf receive-label {
                type rt-types: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 "reroute-mode";
    }
}

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

grouping endpoint-grp {
description "A grouping that defines the structure of " + "an endpoint";
choice ac-or-pw-or-redundancy-grp {
description "A choice of attachment circuit or " + "pseudowire or redundancy group";
case ac {
  description "Attachment circuit(s) as an endpoint";
}
case pw {
  description "Pseudowire(s) as an endpoint";
}
case redundancy-grp {
description "Redundancy group as an endpoint";
choice primary {
    mandatory true;
    description "primary options";
    case primary-ac {
        description "primary-ac";
    }
    case primary-pw {
        description "primary-pw";
    }
}
choice backup {
    description "backup options";
    case backup-ac {
        description "backup-ac";
    }
    case backup-pw {
        description "backup-pw";
    }
}
leaf template {
    type leafref {
        path "/l2vpn/redundancy-group-templates" + 
        "/redundancy-group-template/name";
        description "Reference a redundancy group " + 
        "properties template";
    }
    uses redundancy-group-properties-grp;
}
}

grouping ac-state-grp {
    description "ac-state-grp";
    leaf name {
        type string;
        description "Name of attachment circuit. " + 
        "This field is intended to " + 
        "reference standardized " + 
        "layer-2 definitions.";
    }
    leaf state {
        type operational-state-type;
        description "attachment circuit up/down state";
    }
}
grouping common-pw-state-grp {
    description "common-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 (true) or disabled (false)";
    }
    leaf cw-negotiation {
        type cw-negotiation-type;
        description "Override the control-word negotiation " +
        "preference specified in the " +
        "pseudowire template.";
    }
    leaf tunnel-policy {
        type string;
        description "Used to override the tunnel policy name " +
        "specified in the pseudowire template";
    }
    uses pw-type-grp;
}

grouping vpws-only-pw-state-grp {
    description "vpws-pw-state-grp"
    leaf vccv-ability {
        type boolean;
        description "vccv-ability";
    }
    leaf request-vlanid {
        type uint16;
        description "request vlanid";
    }
    leaf vlan-tpid {
        type string;
        description "vlan tpid";
    }
    leaf ttl {
        type uint8;
    }
description "time-to-live";
});
}
/* L2VPN YANG Model */
container l2vpn {
description "l2vpn";
  container pw-templates {
    description "pw-templates";
    list pw-template {
      key "name";
      description "pw-template";
      leaf name {
        type string;
        description "name";
      }
      leaf mtu {
        type uint16;
        description "pseudowire mtu";
      }
      leaf cw-negotiation {
        type cw-negotiation-type;
        default "preferred";
        description "control-word negotiation preference";
      }
      leaf tunnel-policy {
        type string;
        description "tunnel policy name";
      }
    }
  }
  container redundancy-group-templates {
    description "redundancy group templates";
    list redundancy-group-template {
      key "name";
      description "redundancy-group-template";
      leaf name {
        type string;
        description "name";
      }
      uses redundancy-group-properties-grp;
    }
  }
  container 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 {
   description "BGP auto-discovery parameters";
   leaf route-distinguisher {
      type rt-types:route-distinguisher;
      description "BGP route distinguisher";
   }
   leaf vpn-id {
      type string;
      description "VPN ID";
   }
   uses rt-types:vpn-route-targets;
}
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;
}
list endpoint {
   key "name";
   description "An endpoint";
   leaf name {
      type string;
      description "endpoint name";
   }
   uses endpoint-grp {
      augment "ac-or-pw-or-redundancy-grp/ac" {
         description "Augment for attachment circuit(s) " +
         "as an endpoint";
      }
      list ac {
         key "name";
         leaf name {
            type string;
            description "Name of attachment circuit. " +
            "This field is intended to " +
            "reference standardized " +
            "layer-2 definitions."
         }
      }
   }
}

description "An L2VPN instance’s attachment circuit list";
}

augment "ac-or-pw-or-redundancy-grp/pw" {
    description "Augment for pseudowire(s) as an endpoint";
    list pw {
        key "name";
        leaf name {
            type leafref {
                path ../../../pw/name;
            }
            description "name of pseudowire";
        }
        description "An L2VPN instance’s pseudowire list";
    }
}

augment "ac-or-pw-or-redundancy-grp/redundancy-grp/primary/primary-ac" {
    description "Augment for primary-ac";
    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 "Augment for primary-pw";
    list primary-pw {
        key "name";
        leaf name {
            type leafref {
                path ../../../pw/name;
            }
            description "name of pseudowire";
        }
        description "An L2VPN instance’s pseudowire list";
    }
}

augment "ac-or-pw-or-redundancy-grp/redundancy-grp/backup/backup-ac" {
    description "Augment for backup-ac";
    leaf backup-ac {
        type string;
        description "Name of backup attachment circuit. " +
        "This field is intended to reference standardized layer-2 definitions.";
    }
}
"This field is intended to reference " +
"standardized layer-2 definitions."
}

augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
"backup/backup-pw" {
description "Augment for backup-pw";
list backup-pw {
key "name";
leaf name {

type leafref {
    path "../../../pw/name";
}
description "Reference an attachment circuit";
}
description "A list of backup pseudowires";
}
}
}
}
}
}
}
}
}
container l2vpn-state {
  config false;
  description "l2vpn state";
}
container l2vpn-instances-state {
  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";
      leaf route-distinguisher {
        type rt-types:route-distinguisher;
        description "BGP route distinguisher";
      }
      leaf vpn-id {
        type string;
        description "VPN ID";
      }
      uses rt-types:vpn-route-targets;
    }
    container bgp-signaling {
      description "BGP signaling parameters";
      uses bgp-signaling-parameters-grp;
    }
  }
  container bgp-multipath {
    description "BGP multipath parameters";
    uses bgp-multipath-parameters-grp;
  }
  container bgp-operational {
    description "BGP operational parameters";
    uses bgp-operational-parameters-grp;
  }
}
container l2vpn-instance-state {
  description "L2VPN instance state";
  list l2vpn-instance-state {
    description "An L2VPN instance’s state";
    uses l2vpn-common-parameters-grp;
    container bgp-auto-discovery {
      description "BGP auto-discovery parameters";
      leaf route-distinguisher {
        type rt-types:route-distinguisher;
        description "BGP route distinguisher";
      }
      leaf vpn-id {
        type string;
        description "VPN ID";
      }
      uses rt-types:vpn-route-targets;
    }
    container bgp-signaling {
      description "BGP signaling parameters";
      uses bgp-signaling-parameters-grp;
    }
}
}
}
list endpoint {
    key "name";
    description "An endpoint";
    leaf name {
        type string;
        description "endpoint name";
    }
    uses endpoint-grp {
        augment "ac-or-pw-or-redundancy-grp/ac" {
            description "Augment of attachment circuit state";
            list ac {
                uses ac-state-grp;
                description "An attachment circuit’s " +
                            "operational state";
            }
        }
        augment "ac-or-pw-or-redundancy-grp/pw" {
            description "Augment of pseudowire state";
            list pw {
                uses common-pw-state-grp;
                description "A pseudowire’s operational state";
            }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/primary/ac" {
            description "Augment of primary attachment circuit state";
            container primary-ac {
                uses ac-state-grp;
                description "An attachment circuit’s " +
                            "operational state";
            }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/primary/pw" {
            description "Augment of primary pseudowire state";
            list primary-pw {
                uses common-pw-state-grp;
                description "A pseudowire’s operational state";
            }
        }
        augment "ac-or-pw-or-redundancy-grp/redundancy-grp/backups/ac" {
            description "Augment of backup attachment circuit state";
            container backup-ac {
                uses ac-state-grp;
                description "An attachment circuit’s " +
                            "operational state";
            }
        }
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/backup/backup-pw" {
  description "Augment of backup pseudowire state";
  list backup-pw {
    uses common-pw-state-grp;
    description "A pseudowire’s operational state";
  }
}
/* augments */

augment "/l2vpn/l2vpn-instances/l2vpn-instance" {
  when "type = 'vpws-instance-type'" {
    description "Constraints only for VPWS pseudowires";
  }
  description "Augment for VPWS instance";
  container vpws-constraints {
    must "((count(/endpoint) <= 2) and " +
      "(count(/endpoint/pw) <= 1) and " +
      "(count(/endpoint/ac) <= 1) and " +
      "(count(/endpoint/primary-pw) <= 1) and " +
      "(count(/endpoint/backup-pw) <= 1) " {
      description "A VPWS L2VPN instance has at most 2 endpoints " +
        "and each endpoint has at most 1 pseudowire or " +
        "1 attachment circuit";
    }
    description "VPWS constraints";
  }
}

augment "/l2vpn/l2vpn-instances/l2vpn-instance/pw" {
  when "/../type = 'vpws-instance-type'" {
    description "Pseudowire parameters only for VPWS pseudowires";
  }
  description "Augment for pseudowire parameters for " +
    "VPWS pseudowires";
  leaf vccv-ability {
    type boolean;
    description "vccvability";
  }
  leaf request-vlanid {
    type uint16;
    description "Request VLAN ID";
  }
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 = 'vpls-instance-type'" {
    description "Parameters specifically for a VPLS instance";
  }
  description "Augment for parameters for a VPLS instance";
  uses pbb-parameters-grp;
}

augment "/l2vpn/l2vpn-instances/l2vpn-instance/endpoint" {
  when "./type = 'vpls-instance-type'" {
    description "Endpoint parameter specifically for " + "a VPLS instance";
  }
  description "Augment for endpoint parameters for a VPLS instance";
  leaf split-horizon-group {
    type string;
    description "Identify a split horizon group";
  }
}

augment "/l2vpn/l2vpn-instances/l2vpn-instance/endpoint" + "/ac-or-pw-or-redundancy-grp/redundancy-grp" + "/backup/backup-pw/backup-pw" {
  when "./.type = 'vpls-instance-type'" {
    description "Backup pseudowire parameter specifically for " + "a VPLS instance";
  }
  description "Augment for backup pseudowire parameters for " + "a VPLS instance";
  leaf precedence {
    type uint32;
    description "precedence of the pseudowire";
  }
}

augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" +
"/endpoint/ac-or-pw-or-redundancy-grp/pw/pw" {
  when ".//type = 'vpws-instance-type'" {
    description "Additional operational state specifically for " +
    "a VPWS instance’s pseudowire endpoint";
  }
  description "Augment for a VPWS instance’s pseudowire endpoint " +
  "operational state";
  uses vpws-only-pw-state-grp;
}

augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" +
  "/endpoint/ac-or-pw-or-redundancy-grp/redundancy-grp" +
  "/primary/primary-pw/primary-pw" {
  when ".//type = 'vpws-instance-type'" {
    description "Additional operational state specifically for " +
    "a VPWS instance’s primary pseudowire endpoint";
  }
  description "Augment for a VPWS instance’s primary pseudowire " +
  "endpoint operational state";
  uses vpws-only-pw-state-grp;
}

augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" +
  "/endpoint/ac-or-pw-or-redundancy-grp/redundancy-grp" +
  "/backup/backup-pw/backup-pw" {
  when ".//type = 'vpws-instance-type'" {
    description "Additional operational state specifically for " +
    "a VPWS instance’s backup pseudowire endpoint";
  }
  description "Augment for a VPWS instance’s backup pseudowire " +
  "endpoint operational state";
  uses vpws-only-pw-state-grp;
}

augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" +
  "/endpoint" {
  when ".//type = 'vpls-instance-type'" {
    description "Endpoint parameter specifically for " +
    "a VPLS instance operational state";
  }
  description "Augment for endpoint parameters for a VPLS " +
  "instance operational state";
  leaf split-horizon-group {
    type string;
    description "Identify a split horizon group";
  }
}
augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" {  
    when "type = 'vpls-instance-type'" {  
        description "Additional operational state specifically for " +  
            "a VPLS instance";
    } 
    description "Augment for a VPLS instance’s " +  
        "operational state";
    uses pbb-parameters-state-grp;
}

augment "/l2vpn-state/l2vpn-instances-state/l2vpn-instance" +  
    "/endpoint/ac-or-pw-or-redundancy-grp/redundancy-grp" +  
    "/backup/backup-pw/backup-pw" {  
    when ".//type = 'vpls-instance-type'" {  
        description "Additional operational state specifically for " +  
            "a VPLS instance’s backup pseudowire endpoint";
    } 
    description "Augment for a VPLS instance’s backup pseudowire " +  
        "endpoint operational state";
    leaf precedence {  
        type uint32;  
        description "precedence of the pseudowire";
    }
}

<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


8.2. Informative References


Internet-Draft          YANG Data Model for L2VPN             March 2017


Appendix A. Example Configuration

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

Appendix B. Contributors

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

Reshad Rahman  
Cisco Systems, Inc.  
Email: rrahman@cisco.com

Kamran Raza  
Cisco Systems, Inc.  
Email: skraza@cisco.com

Giles Heron  
Cisco Systems, Inc.  
Email: giheron@cisco.com

Tapraj Singh  
Cisco Systems, Inc.  
Email: tsingh@cisco.com

Zhenbin Li  
Huawei Technologies  
Email: lizhenbin@huawei.com

Zhuang Shunwan  
Huawei Technologies  
Email: Zhuangshunwan@huawei.com

Wang Haibo  
Huawei Technologies
Email: rainsword.wang@huawei.com

Sajjad Ahmed
Ericsson
Email: sajjad.ahmed@ericsson.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

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

Jonathan Hardwick
Metaswitch
Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Nick Delregno
Verizon
Email: nick.deregno@verizon.com

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

Maria Joecylyn
Verizon
Email: joecylyn.malit@verizon.com

Authors' Addresses

Himanshu Shah
Ciena Corporation
Email: hshah@ciena.com

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