L2SM Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2017

B. Wen
Comcast
G. Fioccola
Telecom Italia
C. Xie
China Telecom
L. Jalil
Verizon

October 29, 2016

A YANG Data Model for L2VPN Service Delivery
draft-wen-l2sm-l2vpn-service-model-03

Abstract

This document defines a YANG data model that can be used to configure a Layer 2 Provider Provisioned VPN service.

This model is intended to be instantiated at management system to deliver the overall service. This model is not a configuration model to be used directly on network elements, but provides an abstracted view of the Layer 2 VPN service configuration components. It is up to a management system to take this as an input and use specific configurations models to configure the different network elements to deliver the service. How configuration of network elements is done is out of scope of the document.

The data model in this document includes support for point-to-point Virtual Private Wire Services (VPWS) and multipoint Virtual Private LAN services (VPLS) that use Pseudowires signaled using the Label Distribution Protocol (LDP) and the Border Gateway Protocol (BGP) as described in RFC4761 and RFC6624.

Requirements Language

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

Status of This Memo

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

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

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

Table of Contents

1. Introduction

This document defines a YANG data model for Layer 2 VPN (L2VPN) service configuration. This model is intended to be instantiated at management system to allow a user (a customer or an application) to request the service. This model is not a configuration model to be used directly on network elements, but provides an abstracted view of the L2VPN service configuration components. It is up to a management system to take this as an input and use specific configurations models to configure the different network elements to deliver the service. How configuration of network elements is done is out of scope of the document.

The data model in this document includes support for point-to-point Virtual Private Wire Services (VPWS) and multipoint Virtual Private LAN services (VPLS) that use Pseudowires signaled using the Label Distribution Protocol (LDP) and the Border Gateway Protocol (BGP) as described in [RFC4761] and [RFC6624].

Further discussion of the way that service are modelled in YANG and of the relationship between "customer service models" like the one described in this document and configuration models can be found in [I-D.wu-opsawg-service-model-explained]. Section 4 and Section 6 also provide more information of how this service model could be used and how it fits into the overall modelling architecture.

1.1. Terminology

The following terms are defined in [RFC6241] and are not redefined here:

o client

- configuration data
- server
- state data

The following terms are defined in [RFC6020] and are not redefined here:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC6020].

1.2. Tree diagram

A simplified graphical representation of the data model is presented in Section 5.

The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node and "*" denotes a "list" or "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. Definitions

This document uses the following terms:

Service Provider (SP): The organization (usually a commercial undertaking) responsible for operating the network that offers VPN services to clients and customers.

Customer Edge (CE) Device: Equipment that is dedicated to a particular customer and is directly connected to one or more PE devices via attachment circuits. A CE is usually located at the customer premises, and is usually dedicated to a single VPN, although it may support multiple VPNs if each one has separate attachment circuits. The CE devices can be routers, bridges, switches, or hosts.

Provider Edge (PE) Device: Equipment managed by the SP that can support multiple VPNs for different customers, and is directly connected to one or more CE devices via attachment circuits. A PE is usually located at an SP point of presence (PoP) and is managed by the SP.

Virtual Private LAN Service (VPLS): A VPLS is a provider service that emulates the full functionality of a traditional Local Area Network (LAN). A VPLS makes it possible to interconnect several LAN segments over a packet switched network (PSN) and makes the remote LAN segments behave as one single LAN.

Virtual Private Wire Service (VPWS): A VPWS is a point-to-point circuit (i.e., link) connecting two CE devices. The link is established as a logical through a packet switched network. The CE in the customer network is connected to a PE in the provider network via an Attachment Circuit (AC); the AC is either a physical or a logical circuit. A VPWS differs from a VPLS in that the VPLS is point-to-multipoint, while the VPWS is point-to-point. In some implementations, a set of VPWSs is used to create a multi-site L2VPN network.

3. The Layer 2 VPN Service Model

A Layer 2 VPN service is a collection of sites that are authorized to exchange traffic between each other over a shared infrastructure of a common technology. This Layer 2 VPN service model (L2SM) provides a common understanding of how the corresponding Layer 2 VPN service is to be deployed over the shared infrastructure.

This document presents the L2SM using the YANG data modeling language [RFC6020] as a formal language that is both human-readable and parsable by software for use with protocols such as NETCONF [RFC6241] and RESTCONF [I-D.ietf-netconf-restconf].

This service model is limited to VPWS and VPLS based VPNs as described in [RFC4761] and [RFC6624].

3.1. Applicability of the Layer 2 VPN Service Model

The L2SM defined in this document applies to both point-to-point (E-Line) and multipoint-to-multipoint (E-LAN) carrier Ethernet services.

Over the past decade, The MEF Forum (MEF) has published a series of technical specifications of Ethernet virtual circuit service attributes and implementation agreements between providers. Many Ethernet VPN service providers worldwide have adopted these MEF standards and developed backoffice tools accordingly.

Rather than introducing a new set of terminologies, the L2SM will align with existing MEF attributes when it's applicable. Therefore, service providers can easily integrate any new application that leverages the L2SM data, Service Orchestrator for example, with existing BSS/OSS toolsets. Service providers also have the option to generate L2SM data for current L2VPN customer circuits already deployed in the network.

3.2. Layer 2 VPN Service Types

A Layer 2 VPN circuit can be port-based; in which case any service frames received from subscriber within contractual bandwidth will be delivered to the corresponding remote site, regardless of customer VLAN value (C-tag) of the incoming frame. The service frames can also be native Ethernet frames without C-tag. In this scenario, only one Ethernet Virtual Circuit (EVC) is allowed on a single provider to subscriber link.

Contrary to the above use case, incoming customer service frames may be split into multiple EVCs based on pre-arrangement between the service provider and customer. Typically, C-tag of the incoming frames will serve as the service delimiter for EVC multiplexing over the same provider to subscriber interconnection.

Combining the service-multiplexing attribute with point-to-point verses multipoint-to-multipoint connection type, a Layer 2 VPN circuit may fall under one of the following service types:

- E-Line services: Point-to-Point Layer 2 connections.

  EPL: In its simplest form, a port-based Ethernet Private Line (EPL) service provides a high degree of transparency delivering all customer service frames between UNI-to-UNI interfaces using

All-to-One Bundling. All unicast/broadcast/multicast packets are delivered unconditionally over the EVC. No service multiplexing is allowed on an EPL UNI interface.

EVPL: On the other hand, Ethernet Virtual Private Line (EVPL) service supports multiplexing more than one Point-to-Point, or even other virtual private services, on the same UNI interface. Ingress service frames are conditionally transmitted through one of the EVCs based upon pre-agreed C-tag to EVC mapping. EVPL supports multiple C-tags to one EVC bundling.

- E-LAN services: Multipoint-to-Multipoint Layer 2 connections.

  EP-LAN: Ethernet Private LAN Service (EP-LAN) transparently connects multiple subscriber sites together with All-to-One Bundling. No service multiplexing is allowed on an EP-LAN UNI interface.

  EVP-LAN: Some subscriber may desire more sophisticated control of data access between multiple sites. Ethernet Virtual Private LAN Service (EVP-LAN) allows connecting to multiple EVCs from one or more of the UNI interfaces. Services frame disposition is based on C-tag to EVC mapping. EVP-LAN supports multiple C-tags to one EVC bundling.

3.3. Layer 2 VPN Service Network Topology

Figure 1depicts a typical service provider's physical network topology. Most service providers have deployed an IP, MPLS, or Segment Routing (SR) multi-service core infrastructure. Customer Edge (CE) devices are placed at customer premises as demarcation points to backhaul in profile service frames from the subscriber over the access network to the Provider Edge (PE) equipment. The actual transport technology or physical topology between CE and PE is outside the scope of the L2SM model.

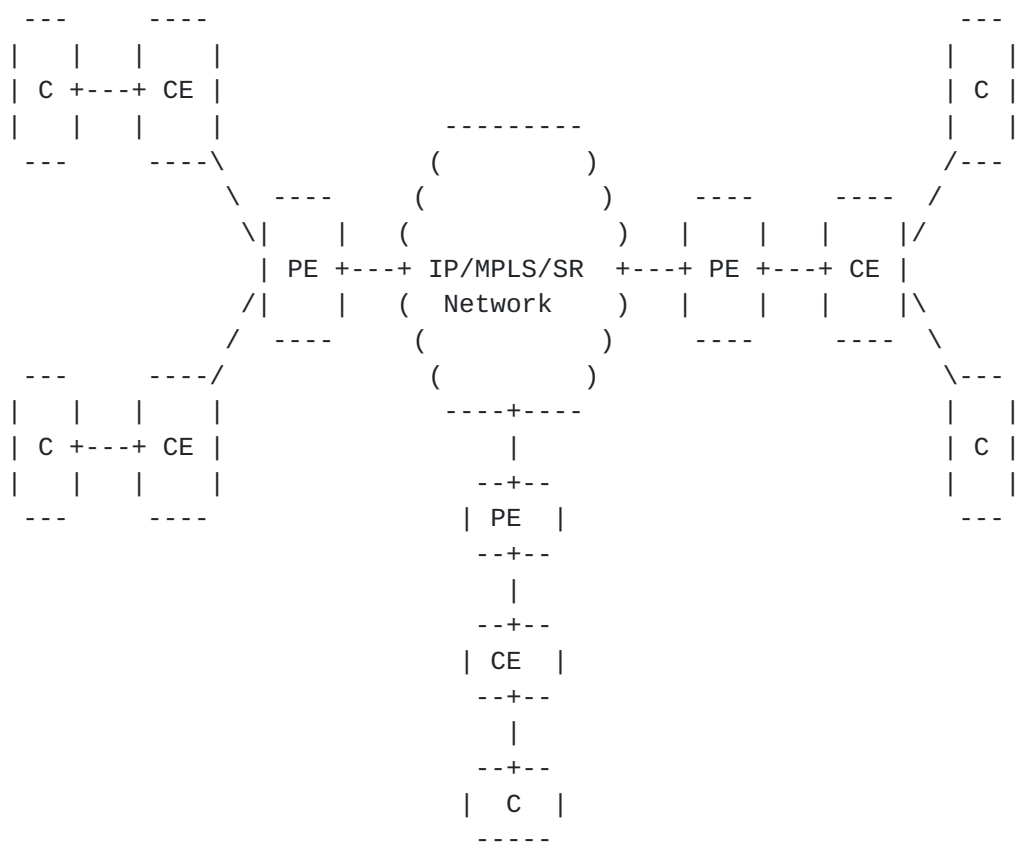

Figure 1: Reference Network for the Use of the L2VPN Service Model

From the subscriber perspective, however, all the edge networks devices are connected over a simulated LAN environment as shown in Figure 2. Broadcast and multicast packets are sent to all participants in the same bridge domain.

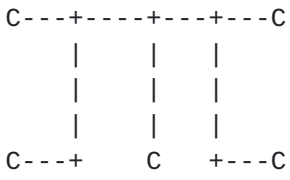

Figure 2: Customer View of the L2VPN

3.4. Layer 2 VPN Ethernet Virtual Circuit Construct

The base model of EVC is shown in Figure 3.

Subscriber edge network device (C) connects to the service provider's CE equipment. The link between C and CE devices is referred as User Network Interface (UNI). For clarification, this is called UNI-C on subscriber side and UNI-N on provider side.

The service provider is obligated to deliver the original service frame across the network to the remote UNI-C. All Ethernet and IP header information, including (but not limit to) source and destination MAC addresses, EtherType, VLAN (C-tag), Class-of-Service marking (802.1p or DSCP), etc.

In coming service frames are first examined at UNI-N based on C-tag, Class-of-Services identifier, EtherType value. Conforming packets are then metered against the contractual service bandwidth. In-profile packets will be delivered to the remote UNI via the Ethernet Virtual Circuit (EVC), which spans between UNI-N to UNI-N.

When both CEs are located in the same provider's network, a single operator maintains the EVC. In this case, the EVC consists only one Operator Virtual Circuit (OVC).

Typically, the CE device at customer premises is a layer 2 Ethernet switch or NID. Service provider may choose to impose an outer VLAN tag (S-tag) into the received subscriber traffic following 802.1ad Q-in-Q standard, especially when Layer 2 aggregation devices exist between CE and PE.

The uplink from CE to PE is referred as Internal Network-to-Network Interface (I-NNI). When 802.1ad Q-in-Q is implemented, Ethernet frames from CE to PE are double tagged with both provider and subscriber VLANs (S-tag, C-tag).

Most service providers have deployed MPLS or SR multi-service core infrastructure. Ingress service frames will be mapped to either Ethernet Pseudowire (PWE) or VxLAN tunnel PE-to-PE. The details of these tunneling mechanism are at the provider's discretion and not part of the L2SM.

Service provider may also choose Seamless MPLS approach to expand the PWE or VxLAN tunnel between UNI-N to UNI-N.

Service provider may leverage multi-protocol BGP to auto discover and signal the PWE or VxLAN tunnel end points.

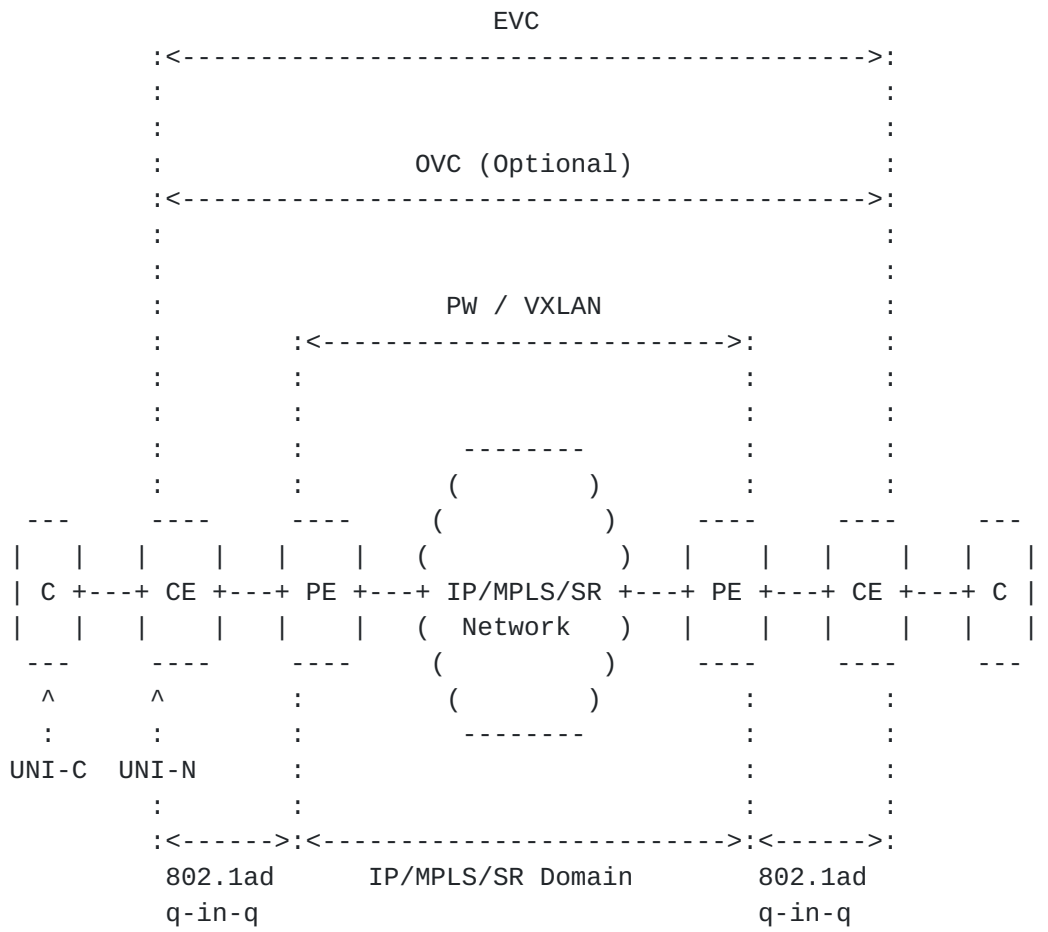

Figure 3: Architectural Model for EVC over a Single Network

Nevertheless, the remote site may be outside of the provider's service territory. In this case, the provider may partner with the operator of another metro network to provider service to the off-net location as shown in Figure 4.

The first provider owns the customer relationship, thus the end-to-end EVC. The EVC is comprised of two or more OVCs. Partially of the EVC is an OVC from local UNI-C to the inter-provider interface. The provider will purchase an Ethernet Access (E-Access) OVC from the second operator to deliver packet to the remote UNI-C.

The inter-connect between the two operators edge gateway (EG) devices is defined as the External Network-to-Network Interface (E-NNI).

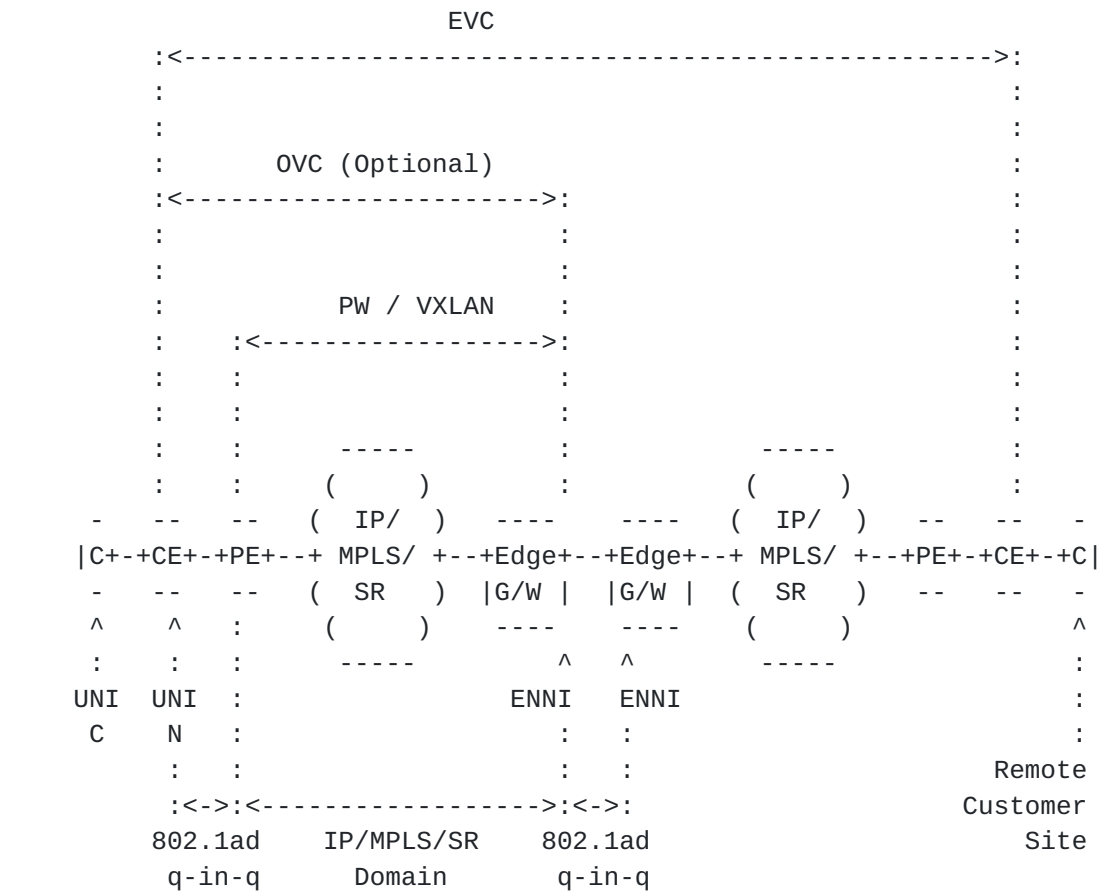

Figure 4: Architectural Model for EVC over Multiple Networks

4. Service Data Model Usage

The L2VPN service model provides an abstracted interface to request, configure and manage the components of a L2VPN service. The model is used by a customer who purchases connectivity and other services from an SP to communicate with that SP.

A typical usage is for this model to be an input for an orchestration layer that is responsible for translating it into configuration commands for the network elements that deliver/enable the service. The network elements may be routers, but also servers (like AAA) necessary within the network.

The configuration of network elements may be done using the Command Line Interface (CLI), or any other configuration (or "southbound") interface such as NETCONF [RFC6241] in combination with device-specific and protocol-specific YANG models.

This way of using the service model is illustrated in Figure 5 and described in more detail in [I-D.wu-opsawg-service-model-explained]. The usage of this service model is not limited to this example: it can be used by any component of the management system, but not directly by network elements.

The usage and structure of this model should be compared to the Layer 3 VPN service model defined in [I-D.ietf-l3sm-l3vpn-service-model].

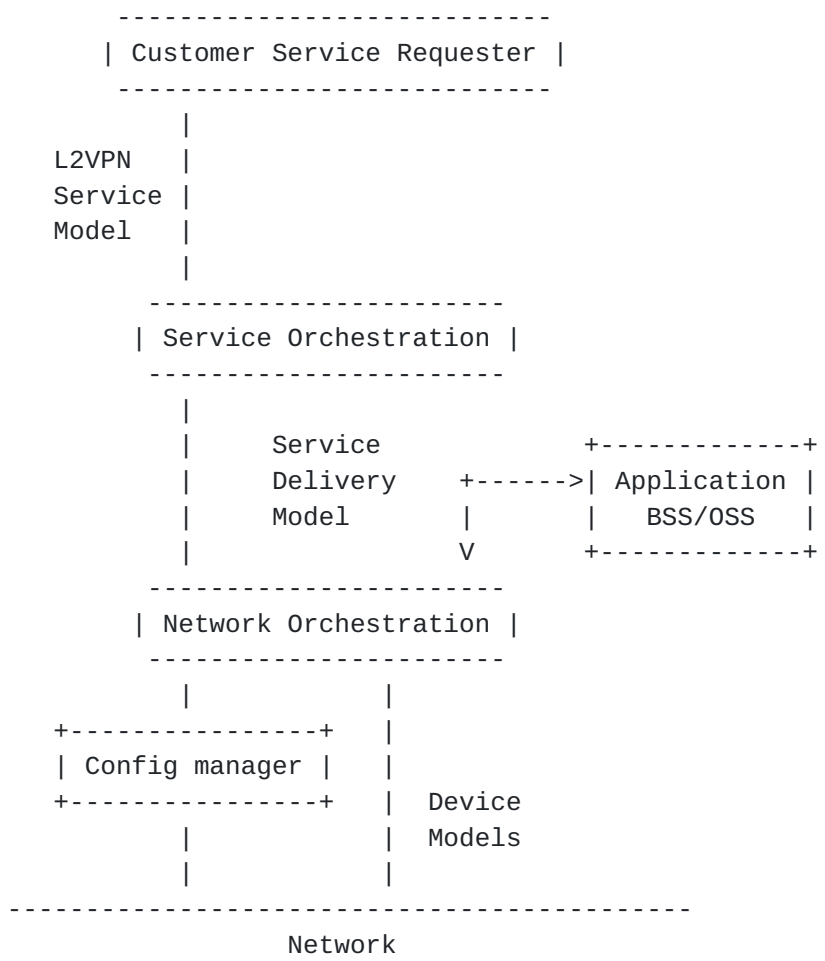

Figure 5: Reference Architecture for the Use of the L2VPN Service Model

Additionally, this data model can be compared with the service delivery models described in [I-D.ietf-bess-l2vpn-yang] and [I-D.ietf-bess-evpn-yang] as discussed in Section 6.

5. Design of the Data Model

The YANG module is divided in three main containers : customer-info, vpn-services, and sites.

The customer-info defines global parameters for a specific customer.

The vpn-svc container under vpn-services defines global parameters for the VPN service for a specific customer.

A site is composed of at least one uni-site or one enni-site.

Authorization of traffic exchange is done through what we call a VPN policy or VPN topology defining routing exchange rules between sites.

The figure below describe the overall structure of the YANG module:

```
module: ietf-l2vpn-svc
+--rw l2vpn-svc
   +--rw customer-info
   |  +--rw customer-info* [customer-account-number customer-name]
   |     +--rw customer-account-number           uint32
   |     +--rw customer-name                     string
   |     +--rw customer-operation-center
   |        +--rw customer-noc-street-address?   string
   |        +--rw customer-noc-phone-number
   |           +--rw main-phone-num?             uint32
   |           +--rw extension-options?          uint32
   +--rw vpn-services
   |  +--rw vpn-svc* [svc-id]
   |     +--rw svc-id                  string
   |     +--rw svc-type
   |     |  +--rw evc
   |     |  |  +--rw evc-id?           boolean
   |     |  |  +--ro number-of-pe?     uint32
   |     |  |  +--ro number-of-site?   uint32
   |     |  +--rw ovc
   |     |     +--rw on-net-ovc-id?   boolean
   |     |     +--rw off-net-ov-id?   boolean
   |     +--rw ethernet-svc-type
   |     |  +--rw (ethernet-svc-type)?
   |     |     +--:(e-line)
   |     |     |  +--rw epl?          boolean
   |     |     |  +--rw evpl?         boolean
   |     |     +--:(e-lan)
   |     |     |  +--rw ep-lan?       boolean
   |     |     |  +--rw evp-lan?      boolean
```

```
|     |     +--:(e-access)
|     |        +--rw access-epl?    boolean
|     |        +--rw access-evpl?   boolean
|     +--rw metro-network-id
|     |  +--rw inter-mkt-service?   boolean
|     |  +--rw intra-mkt* [metro-mkt-id mkt-name]
|     |     +--rw metro-mkt-id      uint32
|     |     +--rw mkt-name          string
|     +--rw signaling-option
|     |  +--rw signaling-option* [name type]
|     |     +--rw name              string
|     |     +--rw type              identityref
|     |     +--rw mp-bgp-l2vpn
|     |     |  +--rw vpn-id?        string
|     |     |  +--rw type?          identityref
|     |     +--rw mp-bgp-evpn
|     |     |  +--rw vpn-id?        string
|     |     |  +--rw type?          identityref
|     |     +--rw t-ldp-pwe
|     |     |  +--rw PE-EG-list* [service-ip-lo-addr vc-id]
|     |     |     +--rw service-ip-lo-addr    inet:ip-address
|     |     |     +--rw vc-id                 string
|     |     +--rw pwe-encapsulation-type
|     |     |  +--rw ethernet?      boolean
|     |     |  +--rw vlan?          boolean
|     |     +--rw pwe-mtu
|     |     |  +--rw allow-mtu-mismatch?      boolean
|     |     +--rw control-word
|     +--rw load-balance-options
|     |  +--rw fat-pw?                boolean
|     |  +--rw entropy-label?         boolean
|     |  +--rw vxlan-source-port?     string
|     +--rw svlan-id-ethernet-tag?    string
|     +--rw cvlan-id-to-evc-map?      string
|     +--rw service-level-mac-limit?  string
|     +--rw service-protection
|     |  +--rw protection-model
|     |  +--rw peer-evc-id
|     +--rw sla-targets
+--rw sites
   +--rw site* [site-id site-type]
      +--rw site-id                     string
      +--rw site-type                   identityref
      +--rw device
      |  +--rw devices* [device-id]
      |     +--rw device-id             string
      |     +--rw site-name?            string
      |     +--rw address?              inet:ip-address
```

```
|     +--rw management-transport?   identityref
+--rw managemnt
|  +--rw type?   identityref
+--rw location
|  +--rw address?        string
|  +--rw zip-code?       string
|  +--rw state?          string
|  +--rw city?           string
|  +--rw country-code?   string
+--rw site-diversity {site-diversity}?
|  +--rw groups
|     +--rw group* [group-id]
|        +--rw group-id  string
+--rw security
+--rw signaling-option {signaling-option}?
|  +--rw signaling-option* [name type]
|     +--rw name         string
|     +--rw type         identityref
|     +--rw mp-bgp-l2vpn
|     |  +--rw vpn-id?   string
|     |  +--rw type?     identityref
|     +--rw mp-bgp-evpn
|     |  +--rw vpn-id?   string
|     |  +--rw type?     identityref
|     +--rw t-ldp-pwe
|     |  +--rw PE-EG-list* [service-ip-lo-addr vc-id]
|     |     +--rw service-ip-lo-addr    inet:ip-address
|     |     +--rw vc-id                 string
|     +--rw pwe-encapsulation-type
|     |  +--rw ethernet? boolean
|     |  +--rw vlan?     boolean
|     +--rw pwe-mtu
|     |  +--rw allow-mtu-mismatch?      boolean
|     +--rw control-word
+--rw load-balance-options
|  +--rw fat-pw?              boolean
|  +--rw entropy-label?       boolean
|  +--rw vxlan-source-port?   string
+--rw ports
   +--rw port* [id]
      +--rw id                      string
      +--rw remote-carrier-name?    string
      +--rw groups
      |  +--rw fate-sharing-group-size?  uint16
      |  +--rw group* [group-id]
      |     +--rw group-id               string
      +--rw bearer
      |  +--rw phy-interface
```

```
|  |  +--rw port-number?           uint32
|  |  +--rw port-speed?            uint32
|  |  +--rw auto-neg?              string
|  |  +--rw phy-mtu?               uint32
|  |  +--rw flow-control?          string
|  |  +--rw encapsulation-type?    enumeration
|  |  +--rw ethertype?             string
|  |  +--rw lldp?                  boolean
|  |  +--rw oam-802.3AH-link {oam-3ah}?
|  |  |  +--rw enable?   boolean
|  |  +--rw uni-loop-prevention?   boolean
|  +--rw LAG-interface
|  |  +--rw LAG-interface* [LAG-interface-number]
|  |     +--rw LAG-interface-number      uint32
|  |     +--rw LACP
|  |        +--rw LACP-state?            identityref
|  |        +--rw LACP-mode?             identityref
|  |        +--rw LACP-speed?            identityref
|  |        +--rw mini-link?             uint32
|  |        +--rw system-priority?       uint16
|  |        +--rw Micro-BFD {Micro-BFD}?
|  |        |  +--rw Micro-BFD-on-off?   enumeration
|  |        |  +--rw bfd-interval?       uint32
|  |        |  +--rw bfd-hold-timer?     uint32
|  |        +--rw bfd {bfd}?
|  |        |  +--rw bfd-enabled?        boolean
|  |        |  +--rw (holdtime)?
|  |        |     +--:(profile)
|  |        |     |  +--rw profile-name? string
|  |        |     +--:(fixed)
|  |        |        +--rw fixed-value?  uint32
|  |        +--rw Member-link-list
|  |        |  +--rw member-link* [name]
|  |        |     +--rw name             string
|  |        |     +--rw port-speed?      uint32
|  |        |     +--rw auto-neg?        string
|  |        |     +--rw mtu?             uint32
|  |        |     +--rw oam-802.3AH-link {oam-3ah}?
|  |        |        +--rw enable?       boolean
|  |        +--rw flow-control?          string
|  |        +--rw encapsulation-type?    enumeration
|  |        +--rw ethertype?             string
|  |        +--rw lldp?                  boolean
|  +--rw interface-description?          string
|  +--rw sub-if-id?                      uint32
+--rw ethernet-connection
|  +--rw vlan
|     +--rw svlan-id-ethernet-tag?       string
```

```
                  +--rw evc-mtu?                          uint32
                  +--rw mac-addr-limit
                  |  +--rw exceeding-option?              uint32
                  +--rw multihoming
                  |  +--rw multihoming* [ESI]
                  |     +--rw ESI                         string
                  |     +--rw (redundancy-mode)?
                  |        +--:(single-active)
                  |        |  +--rw single-active?        boolean
                  |        +--:(all-active)
                  |           +--rw all-active?           boolean
                  +--rw L2CP-control
                  |  +--rw stp-rstp-mstp?            control-mode
                  |  +--rw pause?                    control-mode
                  |  +--rw lacp-lamp?                control-mode
                  |  +--rw link-oam?                 control-mode
                  |  +--rw esmc?                     control-mode
                  |  +--rw l2cp-802.1x?              control-mode
                  |  +--rw e-lmi?                    control-mode
                  |  +--rw lldp?                     boolean
                  |  +--rw ptp-peer-delay?           control-mode
                  |  +--rw garp-mrp?                 control-mode
                  |  +--rw provider-bridge-group?    yang:mac-address
                  |  +--rw provider-bridge-mvrp?     yang:mac-address
                  +--rw service
                  |  +--rw svlan-id-ethernet-tag?    string
                  |  +--rw cvlan-id-to-evc-map?      string
                  |  +--rw service-level-mac-limit?  string
                  |  +--rw service-level
                  |     +--rw cos-identifier?               identityref
                  |     +--rw color-identifier?             identityref
                  |     +--rw ingress-bw-profile-per-evc?   string
                  |     +--rw ingress-bw-profile-per-cos-id? string
                  |     +--rw egress-bw-profile-per-evc?    string
                  |     +--rw egress-bw-profile-per-cos-id? string
                  |     +--rw byte-offset?                  uint16
                  |     +--rw policing?                     identityref
                  |     +--rw performance-tier-option?      identityref
                  |     +--rw COS?                          uint32
                  +--rw B-U-M-strom-control
                  |  +--rw BUM-overall-rate?         uint32
                  |  +--rw BUM-rate-per-type* [type]
                  |     +--rw type                   identityref
                  |     +--rw rate?                  uint32
                  +--rw mac-loop-prevention
                  |  +--rw frequency?                uint32
                  |  +--rw protection-type?          identityref
                  |  +--rw number-retries?           uint32
```

```
+--rw Ethernet-Service-OAM
|  +--rw MD-name?                  string
|  +--rw MD-level?                 uint8
|  +--rw cfm-802.1-ag
|  |  +--rw n2-uni-c* [MAID]
|  |  |  +--rw MAID                      string
|  |  |  +--rw mep-id?                   uint32
|  |  |  +--rw mep-level?                uint32
|  |  |  +--rw mep-up-down?              enumeration
|  |  |  +--rw remote-mep-id?            uint32
|  |  |  +--rw cos-for-cfm-pdus?         uint32
|  |  |  +--rw ccm-interval?             uint32
|  |  |  +--rw ccm-holdtime?             uint32
|  |  |  +--rw alarm-priority-defect?    identityref
|  |  |  +--rw ccm-p-bits-pri?       ccm-priority-type
|  |  +--rw n2-uni-n* [MAID]
|  |     +--rw MAID                      string
|  |     +--rw mep-id?                   uint32
|  |     +--rw mep-level?                uint32
|  |     +--rw mep-up-down?              enumeration
|  |     +--rw remote-mep-id?            uint32
|  |     +--rw cos-for-cfm-pdus?         uint32
|  |     +--rw ccm-interval?             uint32
|  |     +--rw ccm-holdtime?             uint32
|  |     +--rw alarm-priority-defect?    identityref
|  |     +--rw ccm-p-bits-pri?       ccm-priority-type
|  +--rw y-1731* [MAID]
|     +--rw MAID                            string
|     +--rw mep-id?                         uint32
|     +--rw type?                           identityref
|     +--rw remote-mep-id?                  uint32
|     +--rw message-period?                 uint32
|     +--rw measurement-interval?           uint32
|     +--rw cos?                            uint32
|     +--rw loss-measurement?               boolean
|     +--rw synthethic-loss-measurement?    boolean
|     +--rw delay-measurement
|     |  +--rw enable-dm?   boolean
|     |  +--rw two-way?     boolean
|     +--rw frame-size?                     uint32
|     +--rw session-type?                   enumeration
+--rw security
```

Figure 6

5.1. Overview of Main Components of the Model

The L2SM model is structured in a way that allows the provider to list multiple circuits of various service types for the same subscriber.

5.1.1. Customer Information

The "customer-info" container contains essential information to identify the subscriber.

"customer-account-number" is an internal alphanumerical number assigned by the service provider to identify the subscriber. It MUST be unique within the service provider?s OSS/BSS system. The actual format depends on the system tool the provider uses. "customer-name" is in more readable form.

Subscriber operation center and main contact number are also listed here for reference purpose.

5.1.2. VPN Service Overview

The "svc-type" container contains two optional leaves: one for EVC (Ethernet Virtual Connection) and the other one for OVC (Operator Virtual Connection). These two parameters are not mutually exclusive. Depending on the service-type, a Layer 2 VPN service may be identified by EVC-ID, OVC-ID, or both.

E-Line and E-LAN provider shall have EVC-ID assigned to the UNI-to-UNI circuit. If the service has remote UNIs in off-net partner's network, there will be one OVC-ID for the on-net segment between local UNI to the E-NNI interconnect, and one OVC-ID for each off-net segment from E-NNI to the remote UNI. E-Access, on the other hand, is OVC-based service. The E-Access service provider will assign OVC-ID for the circuit between UNI to E-NNI.

The "svc-type" container can be augmented in the future to support other new technologies. Note that the "svc-id" should be corresponding to the "svc-type".

5.1.2.1. Service Type

The "svc-type" container contains two cases, one for EVC (Ethernet Virtual Connection), the other for OVC (Operator Virtual Connection). It can be used to indicate the type of service pipe type. The model user also can augment the "svc-type" container with other cases to support future technologies. Notes that the "svc-id" should be corresponding to the "svc-type".

5.1.2.1.1. EVC

The "evc"case contains an "evc-id" leaf with boolean type. The "evc-id" leaf will be marked TRUE for E-Line and E-LAN service types. And the "svc-id" will be associated with the "evc-id". Only one "evc-id" is allowed for each "svc-id".

The EVC ID is intended to be a structured string. Each service provider can decide the nomenclature in its network. For example, many carriers in North American have implemented the COMMON LANGUAGE? Special Service Circuit Codes (CLCI S/S Codes) - Serial Number Format, which is defined in defined in ANSI ATIS-0300097.

5.1.2.1.2. OVC

The "ovc" case contains two boolean subcases: "on-net-ovc" and "off-net-ovc".

For E-Access or services with off-net UNIs, the "on-net-ovc" leaf MUST be marked TRUE. And the "on-net-ovc-id" will be specified.

In case of E-Access, the "svc-id" will be associated with the "on-net-ovc-id". Only one "on-net-ovc-id" is allowed for each "svc-id".

If the service is E-Line or E-LAN with remote UNIs, there will be one, and only one, "on-net-ovc-id" and a list of "off-net-ovc-id"s for the remote UNIs. However, the "svc-id" is still associated with the "evc-id". Only one "evc-id" is allowed for each "svc-id". New ovc type could be added by augmentation.

5.1.2.2. ethernet-svc-type

The "ethernet-svc-type" group contains all supported Ethernet service types. One, and only one, "ethernet-svc-type" must be selected for each "svc-id".

The current supported Ethernet service types are listed in Section 3.2. New service types can be added in the future.

5.1.2.3. Metro Network Partition

Some service providers may divide their network into multiple administrative domains. And a Layer 2 VPN service may span across more than one metro network of the same service provider. The optional "metro-network-id" container is intended be used by these multi-domain providers to differentiate intra-market versus inter-market services.

When the "inter-mkt-service" leaf is marked TRUE, multiple associated "metro-mkt-id"s will be listed. Otherwise, the service is intra-domain and only one "metro-mkt-id" is allowed.

5.1.2.4. vpn-signaling-option

The "signaling-option" container captures service-wide attributes of the L2VPN instance.

Although topology discovery or network device configurations is purposely out-scoped from the L2SM model, certain VPN parameters are listed here nevertheless. The information here can then be passed to other elements in the whole automation eco-system, such as the configuration engine, which will handle the actual service provisioning function.

The "signaling-option" list uses "name" and "type" combination as the key. The "name" leaf is a free-form string of the VPN instance name. The "type" leaf is for the signaling protocol: BGP-L2VPN, BGP-EVPN, or T-LDP.

5.1.2.4.1. BGP L2VPN

[RFC4761] and [RFC6624] describe the mechanism to auto-discover L2VPN VPLS/VPWS end points (CE-ID or VE-ID) and signal the label base and offset at the same time to allow remote PE to derive the VPN label to be used when sending packets to the advertising router.

Due to the auto-discovery natural, PEs that have at least one attachment circuit associated with a particular VPN service do not need to be specified explicitly.

In the L2SM model, only the target community (or communities) will be listed at the service level.

The "type" leaf under "mp-bgp-l2vpn" is an identityref to specify "vpws" or "vpls" sub-types.

5.1.2.4.2. BGP EVPN

Defined in [RFC7432], EVPN is a new promising L2VPN technology based upon BGP MAC routing. It's considered the next generation L2VPN solution that provides similar functionality of BGP VPWS/VPLS with improvement around redundancy, multicast optimization, provisioning and simplicity.

Due to the auto-discovery natural, PEs that have at least one attachment circuit associated with a particular VPN service do not need to be specified explicitly.

In the L2SM model, only the target community (or communities) will be listed at the service level.

The "type" leaf under "mp-bgp-evpn" is an identityref to specify "vpws" or "vpls" sub-types.

5.1.2.4.3. LDP Pseudowires

[RFC4762] specified the method of using targeted LDP sessions between PEs to exchange VC label information. This requires a manually define a full mesh of targeted LDP sessions between all PEs.

As multiple attachment circuits may terminate on a single PE, this PE-to-PE mesh is not a per site attribute. All PEs related to the L2VPN service will be listed in the "t-ldp-pwe" with associated "vc-id".

5.1.2.4.4. PWE Encapsulation Type

Based on [RFC4448], there are two types of Ethernet services: "Port-to-Port Ethernet PW emulation" and "Vlan-to-Vlan Ethernet PW emulation", commonly referred to as Type 5 and Type 4 respectively. This concept applies to both BGP L2VPN VPWS/VPLS and T-LDP signaled PWE implementations.

The "pwe-encapsulation-type" container contains two Boolean type leaves: "ethernet" and "ethernet-vlan", only one should be marked TRUE if "signaling-option" is "mp-bgp-l2vpn" or "t-ldp-pwe".

5.1.2.4.5. PWE MTU

During the signaling process of BGP-L2VPN or T-LDP pseudowire, the pwe-mtu value is exchanged and must match on both ends. By default, the pwe-mtu is derived from physical interface MTU of the attachment circuit minus the EoMPLS transport header. In some cases, however, the physical interface on both ends of the circuits may not have identical MTU settings. For example, due to 802.1ad q-in-q operation, I-NNI interface will need extra four bytes to accommodate the S-tag. The inter-carrier E-NNI link may also have a different MTU size then the internal network interfaces.

[RFC4448] requires same MTU size on physical interface on both end of the pseudowire. In actual implementations, many router vendors have

provided the knob to explicitly specify the pwe-mtu, which can then be decoupled from the physical interface MTU.

When there's a mismatch between the physical interface MTU and configured pwe-mtu, "allow-mtu-mismatch" knob is also required in many cases.

The optional "pwe-mtu" container is for this purpose.

5.1.2.4.6. Control Word

A control word is an optional 4-byte field located between the MPLS label stack and the Layer 2 payload in the pseudowire packet. It plays a vital role in Any Transport over MPLS (AToM). The 32-bit field carries generic and Layer 2 payload-specific information, including a C-bit which indicates whether the control word will present in the Ethernet over MPLS (EoMPLS) packets. If the C-bit is set to 1, the advertising PE expects the control word to be present in every pseudowire packet on the pseudowire that is being signaled. If the C-bit is set to 0, no control word is expected to be present.

Whether to include control word in the pseudowire packets MUST match on PEs at both ends of the pseudowire and it's non-negotiable during the signaling process.

Control-word applies to both BGP L2VPN VPWS/VPLS and T-LDP signaled PWE implementations. It is a routing-instance level configuration in many cases.

The optional "control-word" leaf is a Boolean field in the L2SM model for the provider to explicitly specify whether control-word will be signaled for the service instance.

5.1.2.5. Load Balance Option

As the subscribers start to deploy more 10G or 100G Ethernet equipment in their network, the demand for high bandwidth Ethernet services increases. Along with the great revenue opportunities, these high bandwidth service requests also pose challenges on capacity planning and service delivery in the provider's network. Especially when the contractual bandwidth is at, or close to, the speed of physical link of the service provider's core network. Because of the encapsulation overhead, the provider can not deliver the throughput in the service level agreement over a single link. Although there may be bundled Nx10G or Nx100G aggregation links between core network elements, or Equal Cost Multiple Paths (ECMP) in the network, an EoMPLS PWE or VxLAN circuit is considered a single

flow to a router or switch which uses the five tuples in the hashing algorithm.

Without burdening the core routers with additional processing of deep inspection into the payload, the service provider now have the option of inserting flow or entropy label into the EoMPLS frames, or using different source UDP ports in case of VxLAN/EVPN, at ingress PE to facility load-balancing on the subsequent nodes along the path. The ingress PE is in a unique position to see the actual unencapsulated service frames and identify data flows based on the original Ethernet and IP header.

On the other hand, not all Layer 2 Ethernet VPNs is suited for load-balancing across diverse ECMP paths. For example, a Layer 2 Ethernet service transported over a single RSVP signaled LSP will not take multiple ECMP paths. Or if the subscriber is concerned about latency/jitter then diverse path load-balance can be undesirable.

The optional "load-balance-option" container is intended to capture the load-balance agreement between the subscriber and provider. If the "load-balance" Boolean leaf is marked TRUE, then one of the following load-balance methods can be selected: "fat-pw", "entropy-label", or "vxland-source-udp-port".

5.1.2.6. SVLAN ID Ethernet Tag

Service providers have the option of inserting an outer VLAN tag (the S-tag) into the service frames from the subscriber to improve service scalability and customer VLAN transparency.

Ideally, all external interfaces (UNI and E-NNI) associated with a given service will have the same S-tag assigned. However, this may not always be the case. Traffic with all attachments using different S-tags will need to be "normalized" to a single service S-tag. (One example of this is a multipoint service involves multiple off-net OVCs terminating on the same E-NNI interface. Each of these off-net OVCs will have a distinct S-tag, which can be different from the S-tag used in the on-net part of the service.)

The purpose of the optional "svlan-id-ethernet-tag" leaf is to identify the service-wide "normalized S-tag".

5.1.2.7. CVLAN ID To EVC MAP

When more than one services are multiplexed on the same interface, ingress service frames are conditionally transmitted through one of the EVC/OVCs based upon pre-arranged customer VLAN to EVC mapping. Multiple customer VLANs can be bundled across the same EVC.

"cvlan-id-to-evc-map", when applicable, contains the list of customer vlans to the service mapping in a free-form format. In most cases, this will be the VLAN access-list for the inner 802.1q tags (the C-tag).

5.1.2.8. Service Level MAC Limit

When multiple services are provided on the same network element, MAC address table, and RIB space for MAC-routes in case of EVPN, are shared common resource. Service providers may impose a maximum number of MAC learned from the subscriber for a single service instance, and specify the action when the upper limit is violated: drop the packet, flood the packet, or simply send a warning log message.

For point-to-point services, if MAC learning is disabled then MAC limit is not necessary in this kind of implementation.

The optional "service-level-mac-limit" container contains the subscriber MAC address limit and exceeding action information.

5.1.2.9. Service Protection

Sometimes the subscriber may desire end-to-end protection at the service level for applications with high availability requirements. There are two protection schemes to offer redundant services:

- 1+1 protection: In this scheme, the primary EVC or OVC will be protected by a backup EVC or OVC, typically meet certain diversified path/fiber/site/node criteria. Both primary and protection circuits are provisioning to be in forwarding state. Subscriber may choose to send the same service frames across both circuits simultaneously.

- 1:1 protection: In this scheme, a backup circuit can be provisioning to the primary circuits. Depending on the implementation agreement, the protection circuits may either always be in forwarding state, or only become active when detecting a faulty state or the primary circuit.

The optional "service-protection" container hereby is to capture the desired service protection agreement between subscriber and provider.

An "peer-evc-id" should be specified when the "protection-model" has value.

5.1.3. site

The "site" container is intended for the provider to store information of detailed implementation arrangement with either the subscriber or peer operators at each inter-connect location.

We are restricting the L2SM to exterior interfaces only. All internal interfaces or the underlying topology is outside the scope of L2SM.

There are possibly two types of external facing connections associated with an Ethernet VPN service:

- UNI site: where a customer edge device connects to one or more VPN services.

- E-NNI site: where two Ethernet service providers inter-connect with each other.

Most of the attributes of a site are common to the two typs of site and so are presented just once. Divergences, that is, attributes that are specific to the type of site, are captured in type-dependent containers.

For each site, there are sub-containers to maintain physical link attributes, service frame and Layer 2 control protocol frame disposition, Ethernet service OAM attributes, and service bandwidth profile and priority level agreement.

5.1.3.1. Generic Site Objects

Typically, the following characteristics of a site interface handoff need to be documented as part of the service design:

Unique identifier (site-id) : An arbitrary string to uniquely identify the site within the overall network infrastructure. The format of site-id is determined by the local administration of the VPN service.

Site Type (site-type) : TBD.

Device (device) : TBD.

Management (management) : Defines the model of management of the site, for example: type, management-transport, address.

Location (location) : The site location information to allow easy retrieval on nearest available resources.

Site diversity (site-diversity) : Presents some parameters to support site diversity.

Site security (security) : TBD.

Site signaling (signaling-options) : TBD.

Load balancing (load-balance-options) : TBD.

Ports (ports) : Defines the list of ports to the sites and their properties.

5.1.3.1.1. Site ID

The "site-id" leaf contains an arbitrary string to uniquely identify the site within the overall network infrastructure. The format of the site-id is determined by the local administration of the VPN service.

5.1.3.1.2. Site Management

The "management" sub-container is intended for site management options, depending on the device ownership and security access control. The followings are three common management models:

CE Provider Managed : The provider has the sole ownership of the CE device. Only the provider has access to the CE. The responsibility boundary between SP and customer is between CE and customer network. This is the most common use case.

CE Customer Managed : The customer has the sole ownership of the CE device. Only the customer has access to the CE. In this model, the responsibility boundary between SP and customer is between PE and CE.

CE Co-managed : The provider has ownership of the CE device and responsible for managing the CE. However, the provider grant the customer accessing the CE for some configuration/monitoring purpose. In this co-managed mode the responsibility boundary is the same as the provider-managed model.

The selected management mode is specified under the "type" leaf. The "address" leaf stores CE device management IP information. And " management-transport" leaf is used to identify the transport protocol for management traffic, IPv4 or IPv6. Additional security options MAY be derived based on the particular management model selected.

5.1.3.1.3. Site Location

The information in the "location" sub-container under a "site" allows easy retrieval on nearest available facility for access topology planning. It may also be used by other network orchestration component to decide the targeted upstream PE. Location is express in terms of postal information.

5.1.3.1.4. Site Diversity

Some subscriber may request upstream PE diversity between two or more sites. These sites will share the same diversity group ID under the optional "site-diversity" sub-container.

5.1.3.1.5. Site Security

This sub-container is is a placeholder for site-security options. It presents parameters for ingress service stream admission control and encryption profile information.

5.1.3.1.6. Site Signaling Option

See Section 5.1.2.4.

5.1.3.1.7. Site Load Balance Options

TBD.

5.1.3.1.8. Ports

The L2SM includes a set of essential physical interface properties and Ethernet layer characteristics in the "port" sub-container. Some of these are critical implementation arrangements that require consent from both subscriber and provider.

5.1.3.1.8.1. ID

"id" is a free-form string to identify a given interface. Service provider can decide on the actual nomenclature used in the management systems.

5.1.3.1.8.2. Remote Carrier Name

TBD.

5.1.3.1.8.3. Fate Sharing Group

TBD.

5.1.3.1.8.4. Bearer

Under port, there is a bearer container that presents two sets of link attributes: physical or optional LAG interface attributes. These parameters are essential for the connection between subscriber and provider edge devices to establish properly.

For each physical interface (phy-interface), there are basic configuration parameters like port number and speed, interface face MTU, auto-negotiation and flow-control settings, etc. "encapsulation-type" is for user to select between Ethernet encapsulation (port-based) or Ethernet VLAN encapsulation (VLAN-based). All allowed Ethertypes of ingress service frames can be listed under "ethertype". In addition, the subscriber and provider may decide to enable advanced features, such as LLDP, 802.3AH link OAM, MAC loop detection/prevention at a UNI, based on mutual agreement.

Sometimes the subscriber may require multiple physical links bundled together to form single logical point-to-point LAG connection to the service provider. Typically, LACP (Link Aggregation Control Protocol) is used to dynamically manage adding or deleting member links of the aggregate group. In general, LAG allows for increased service bandwidth beyond the speed of a single physical link while providing graceful degradation as failure occurs, thus increased availability.

In the L2SM, there is a set of attributes under "LAG-interface" related to link aggregation functionality. The subscriber and provider first need to decide on whether LACP PDU will be exchanged between the edge device by specifying the "LACP-state" to "On" or "Off". If LACP is to be enabled, then both parties need to further specify whether it will be running in active versus passive mode, plus the time interval and priority level of the LACP PDU. Subscriber and provider can also determine the minimum aggregate bandwidth for a LAG group to be considered valid path by specifying the optional "mini-link" attribute. To enable fast detection of faulty links, Micro-BFD runs independent UDP sessions to monitor the status of each member link. Subscriber and provider should consent to the BFD hello interval and hold time.

Each member link will be listed under the LAG interface with basic physical link properties. Certain attributes like flow-control,

encapsulation type, allowed ingress Ethertype and LLDP settings are at the LAG level.

If the Ethernet service is enabled on a logical unit on the connection at the interface, the "sub-if-id" should be specified.

5.1.3.1.8.5. Ethernet Connection

The "Ethernet-connection" container presents site specific (S-tag, C-tag) management options. The overall S-tag for the Ethernet circuit and C-tag to EVC mapping, if applicable, has been placed in the service container. The S-tag under "port" should match the S-tag in service container in most cases. However, vlan translation is required for the S-tag in certain deployment at the external facing interface or upstream PEs to "normalize" the outer VLAN tag to the service S-tag into the network and translate back to the site' S-tag in the opposite direction. One example of this is, with Layer 2 aggregation switch alone the path, the S-tag for the EVC has been previously assigned to another service thus can not be used by this attachment circuit. Another use case is when multiple E-access OVCs from the same E-NNI interfaces are attached to the same E-LAN service.

The "svlan-id-ethernet-tag" in the "Ethernet-connection" container is either the S-tag inserted at a UNI or the outer tag of ingress packets at an E-NNI. These parameters are included in the L2SM to facilitate other management system to generate proper configuration for the network elements.

"Ethernet-connection" container also contains optional site-specific C-tag to EVC mapping.

5.1.3.1.8.6. EVC MTU

The maximum MTU of subscriber service frames can be derived from the physical interface MTU by default, or specified under the "evc-mtu" leaf if it is different than the default number.

5.1.3.1.8.7. MAC Address Limit

The service provider may choose to impose a per attachment circuit "mac-addr-limit" in addition to the service-lever MAC limit, and specify the exceeding options accordingly.

5.1.3.1.8.8. Multihoming

EVPN supports PE geo-redundancy in the access domain. The connection between a multi-homed CE to PE is identified with a uniquely assigned ID referred as an Ethernet Segment Identifier (ESI). Because a learned MAC address is propagated via BGP, it allows for multiple active paths in forwarding state and load-balancing options.

The "multihoming" container contains ESI and redundancy mode attribute for EVPN multi-homing site.

5.1.3.1.8.9. L2CP-Control

To facilitate interoperability between different MSOs, MEF has provided normative guidance on Layer 2 control protocol (L2CP) processing requirements for each service type. Subscriber and provider should make pre-arrangement on whether to allow interaction between the edge device or keep each others control plane separate on a per protocol base.

The destination MAC addresses of these L2CP PDUs fall within two reserved blocks specified by IEEE 802.1 Working Group. Packet with destination MAC in these multicast ranges has special forwarding rules.

- Bridge Block of Protocols: 01-80-C2-00-00-00 through 01-80-C2-00-00-0F

- MRP Block of Protocols: 01-80-C2-00-00-20 through 01-80-C2-00-00-2F

Layer 2 protocol tunneling allows service providers to pass subscriber Layer 2 control PDUs across the network without being interpreted and processed by intermediate network devices. These L2CP PDUs are transparently encapsulated across the MPLS-enabled core network in Q-in-Q fashion.

The "L2CP-control" container contains the list of commonly used L2CP protocols. Service provider can specify DISCARD, PEER or TUNNEL action for each individual protocol.

In addition, "provider-bridge-group" and "provider-bridge-mvrp" addresses are also listed in the L2CP container.

5.1.3.1.8.10. Service Profile

In MEF 23.2 ([MEF-23-2]) two types of model are defined as the following:

Class-of-Service Identifier based on EVC or OVC EP (End Point): In this model, regardless of customer marking, all in-profile frames will be marked with service level in contractual agreement. Customer CoS markings are preserved throughout the provider network. The bandwidth profile consists of one set of CIR/CBS and EIR/EBS values.

Class-of-Service Identifier based on Priority Code Point: Using this model, multiple classes of services can be associated with a single customer EVC, identified by dot1p bits in the C-tag. Each service level has its own individual bandwidth profile. Out-of-profile packets will be discarded. Customer CoS markings are preserved.

Class-of-Service Identifier based on DSCP: Using this model, multiple classes of services can be associated with a single customer EVC, identified by DSCP bits in the IP header. Each service level has its own individual bandwidth profile. Out-of-profile packets will be discarded. Customer CoS markings are preserved.

Similarly, Color-identifier can be assigned based on EVC or OVC EP, dot1p value in C-tag, or DSCP in IP header. Ingress service frames are metered against the bandwidth profile based on the cos-identifier. A "color" will be assigned to a service frame to identify its bandwidth profile conformance. A service frame is "green" if it is conformant with "committed" rate of the bandwidth profile. A Service Frame is "yellow" if it is exceeding the "committed" rate but conformant with the "excess" rate of the bandwidth profile. Finally, a service frame is "red" if it is conformant with neither the "committed" nor "excess" rates of the bandwidth profile.

Ingress/egress-bandwidth-profile-per-evc presents the ingress/egress bandwidth profile per EVC, providing rate enforcement for all ingress service frames at the interface that are associated with a particular EVC.

Alternately, ingress/egress-bandwidth-profile-per-cos-id presents the ingress/egress bandwidth profile per CoS, providing rate enforcement for all service frames for a given class of service. The class of service is identified via a CoS identifier. So this bandwidth profile applies to service frames over an EVC with a particular CoS

value. Multiple ingress/egress-bandwidth-profile-per-cos-id can be associated with the same EVC.

The optional "byte-offset" indicates how many bytes in the service frame header are excluded from rate enforcement.

5.1.3.1.8.11. BUM Strom Control

For point-to-point E-LINE services, the provider only needs to deliver a single copy of each service frame to the remote PE, regardless whether the destination MAC address of the incoming frame is unicast, multicast or broadcast. Therefore, all in-profile service frames should be delivered unconditionally.

B-U-M (Broadcast-UnknownUnicast-Multicast) frame forwarding in multipoint-to-multipoint services, on the other hand, involves both local flooding to other attachment circuits on the same PE and remote replication to all other PEs, thus consumes additional resources and core bandwidth. Special B-U-M frame disposition rules can be implemented at external facing interfaces (UNI or E-NNI) to rate-limit the B-U-M frames, in term of number of packets per second or bits per second.

The threshold can apply to all B-U-M traffic, or one for each category.

5.1.3.1.8.12. MAC Loop Protection

MAC address flapping between different physical ports typically indicates a bridge loop condition in the subscriber network. Misleading entries in the MAC cache table can cause service frames to circulate around the network indefinitely and saturate the links throughout the provider's network, affecting other services in the same network. In case of EVPN, it also introduces massive BGP updates and control plane instability.

Service provider may opt to implement switching loop prevention mechanism at the external facing interfaces for multipoint-to-multipoint services by imposing a MAC address move threshold.

The MAC move rate and prevention-type options are listed in the "mac-loop-prevention" container.

5.1.3.1.8.13. Ethernet Service OAM

The advent of Ethernet as wide-area network technology brings additional requirements of end-to-end service monitoring and fault management in the carrier network, particularly in the area of

service availability and Mean Time To Repair (MTTR). Ethernet Service OAM in the L2SM refers to the combined protocol suites of IEEE 802.1ag ([IEEE-802-1ag]) and ITU-T Y.1731 ([ITU-T-Y-1731]).

Generally speaking, Ethernet Service OAM enables service provider to perform service continuity check, fault-isolation, and packet delay/jitter measurement at per customer per EVC granularity. The information collected from Ethernet Service OAM data sets is complementary to other higher layer IP/MPLS OSS tools to ensure the required service level agreements (SLAs) can be meet.

The 802.1ag Connectivity Fault Management (CFM) functional model is structured with hierarchical maintenance domains (MD), each assigned a unique maintenance level. Higher level MD can be nested over lower level MD. However, the MD can not intersect. The scope of each MD can be solely within a subscriber's network, solely within the provider's network, interact between the subscriber-to-provider or provider-to-provider edge equipment, or tunnel over another provider's network.

Depending on the use case scenario, one or more maintenance end point (MEP) can be placed on the external facing interface, sending CFM PDUs towards the core network (UP MEP) or downstream link (DOWN MEP).

The "cfm-802.1-ag" sub-container under "port" currently presents two types of CFM maintenance association (MA): UP MEP for UNI-N to UNI-N MA and DOWN MEP for UNI-N to UNI-C MA. For each MA, user can define the maintenance domain ID (MAID), MEP level, MEP direction, remote MEP ID, CoS level of the CFM PDUs, Continuity Check Message (CCM) interval and hold time, alarm priority defect, CCM priority-type, etc.

ITU-T Y.1731 Performance Monitoring (PM) provides essential network telemetry information that includes the measurement of Ethernet service frame delay, frame delay variation, frame loss, and frame throughput. The delay/jitter measurement can be either one-way or two-way. Typically, a Y.1731 PM probe sends a small amount of synthetic frames along with service frames to measure the SLA parameters.

The "y-1731" sub-container under "port" contains a set of parameters for use to define the PM probe information, including MAID, local and remote MEP-ID, PM PDU type, message period and measurement interval, CoS level of the PM PDUs, loss measurement by synthetic or service frame options, one-way or two-way delay measurement, PM frame size, and session type.

6. Interaction with Other YANG Modules

As expressed in Section 4, this service module is not intended to configure the network element, but is instantiated in a management system.

The management system might follow modular design and comprise at least two different components:

a. The component instantiating the service model (let' call it the service component)

b. The component responsible for network element configuration (let's call it the configuration component)

In some cases when the split is needed between the behavior and functions that a customer requests and the technology that the network operator has available to deliver the service [I-D.wu-opsawg-service-model-explained]. A new component can be separated out of the service component (let's call it the control component). This component is responsible for network-centric operation and is aware of many features such as topology, technology, and operator policy. As an optional component, it can use service model as input and is not required if the control component delegates its control operations to the configuration component.

In Section 7 we provide some example of translation of service provisioning request to router configuration lines as an illustration. In the NETCONF/YANG ecosystem, it is expected that NETCONF and YANG will be used between the configuration component and network elements to configure the requested service on those elements.

In this framework, it is expected that YANG models will be used for configuring service components on network elements. There will be a strong relationship between the abstracted view provided by this service model and the detailed configuration view that will be provided by specific configuration models for network elements such as those defined in [I-D.ietf-bess-l2vpn-yang] and [I-D.ietf-bess-evpn-yang]. Service components needing configuration on network elements in support of the service model defined in this document include:

- VRF definition including VPN policy expression.
- Physical interface.
- Ethernet layer (VLAN ID).

- QoS : classification, profiles, etc.
- Routing protocols : support of configuration of all protocols listed in the document, as well as routing policies associated with these protocols.
- Multicast Support.
- Ethernet Service OAM Support.

7. Service Model Usage Example

*** TBD ***

8. YANG Module

```
<CODE BEGINS>
file "ietf-l2vpn-svc@2016-09-29.yang"
module ietf-l2vpn-svc {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc";
  prefix "l2svc";
  import ietf-inet-types {
    prefix inet;
  }
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IETF L2SM Working Group.";
  contact
    "WG List: l2sm@ietf.org
     Editor: Bin_Wen@comcast.com";
  description
    "The YANG module defines a generic service configuration
     model for Layer 2 VPN services common across all of the
     vendor implementations.";
  revision 2016-10-15 {
    description
      "Initial revision.";
    reference
      "draft-wen-l2sm-l2vpn-service-model-01.txt
       A YANG Data Model for L2VPN Service Delivery.";
  }

  /* Features */

  feature oam-3ah {
```

```
      description
        "Enables support of OAM 802.3ah";
    }

    feature Micro-BFD {
      description
        "Enables support of Micro-BFD";
    }

    feature bfd {
      description
        "Enables support of BFD";
    }

    feature signaling-option {
      description
        "Enable support of signaling option";
    }

    feature site-diversity {
      description
        "Enables support of site diversity constraints";
    }

    feature encryption {
      description
        "Enables support of encryption";
    }

    /* Typedefs */

    typedef ccm-priority-type {
      type uint8 {
        range "0..7";
      }
      description
        "A 3 bit priority value to be used in the VLAN tag, if present
         in the transmitted frame.";
    }
    typedef control-mode {
      type enumeration {
        enum peer {
          description
            "Peer mode";
        }
        enum tunnel {
          description
            "Tunnel mode";
```

```
        }
        enum discard {
          description
            "Discard mode";
        }
      }
      description
        "Defining a type of the control mode";
    }

    /* Identities */

    identity site-type {
      description
        "Identity of site type.";
    }

    identity uni {
      base site-type;
      description
        "Identity of User Network Interface ";
    }

    identity enni {
     base site-type;
     description
       "Identity of External Network to Network Interface";
    }

    identity color-id {
      description
        "Identity of color id";
    }

    identity color-id-evc {
      base color-id;
      description
        "Identity of color id base on EVC";
    }

    identity color-id-evc-cvlan {
      base color-id;
      description
        "Identity of color id base on EVC and CVLAN ";
    }

    identity cos-id {
      description
```

```
        "Identity of class of service id";
    }

    identity cos-id-evc {
      base cos-id;
      description
        "Identity of cos id based on EVC";
    }

    identity cos-id-evc-pcp {
      base cos-id;
      description
        "Identity of cos id based on EVC and PCP";
    }

    identity cos-id-evc-dscp {
      base cos-id;
      description
        "Identity of cos id based on EVC and DSCP";
    }

    identity cos-id-ovc-ep {
      base cos-id;
      description
        "Identity of cos id based on OVC EP";
    }

    identity performance-tier-option {
      description
        "Identity of performance tier option.";
    }

    identity metro {
      base performance-tier-option;
      description
        "Identity of metro";
    }

    identity regional {
      base performance-tier-option;
      description
        "Identity of regional";
    }

    identity continental {
      base performance-tier-option;
      description
        "Identity of continental";
```

```
    }

    identity global {
      base performance-tier-option;
      description
        "Identity of global";
    }

    identity policing {
      description
        "Identity of policing type";
    }

    identity one-rate-two-color {
      base policing;
      description
        "Identity of one-rate, two-color (1R2C)";
    }

    identity two-rate-three-color {
      base policing;
      description
        "Identity of two-rate, three-color (2R3C)";
    }

    identity BUM-type {
      description
        "Identity of BUM type";
    }

    identity broadcast {
      base BUM-type;
      description
        "Identity of broadcast";
    }

    identity unicast {
      base BUM-type;
      description
        "Identity of unicast";
    }

    identity multicast {
      base BUM-type;
      description
        "Identity of multicast";
    }
```

```
identity loop-prevention-type{
  description
    "Identity of loop prevention";
}

identity shut {
  base loop-prevention-type;
  description
    "Identity of shut protection";
}

identity trap {
  base loop-prevention-type;
  description
    "Identity of trap protection";
}

identity lacp-state {
  description
    "Identity of LACP state";
}

identity lacp-on {
  base lacp-state;
  description
    "Identity of LCAP on";
}

identity lacp-off {
  base lacp-state;
  description
    "Identity of LACP off";
}

identity lacp-mode {
  description
    "Identity of LACP mode";
}

identity lacp-passive {
  base lacp-mode;
  description
    "Identity of LACP passive";
}

identity lacp-active {
  base lacp-mode;
  description
```

```
        "Identity of LACP active";
    }

    identity lacp-speed {
      description
        "Identity of LACP speed";
    }

    identity lacp-fast {
      base lacp-speed;
      description
        "Identity of LACP fast";
    }

    identity lacp-slow {
      base lacp-speed;
      description
        "Identity of LACP slow";
    }

    identity vpn-signaling-type {
       description
         "Identity of VPN signaling types";
    }

    identity vrf {
      base vpn-signaling-type;
      description
        "Virtual routing and forwarding (VRF).";
    }

    identity vfi {
      base vpn-signaling-type;
      description
        "Virtual forwarder interface";
    }

    identity evi {
      base vpn-signaling-type;
      description
        "Ethernet virtual interconnect.";
    }

    identity l2vpn-type {
      description
        "Layer 2 VPN types";
    }
```

```
identity vpws {
  base l2vpn-type;
  description
    "Virtual Private Wire Service";
}

identity vpls {
  base l2vpn-type;
  description
    "Virtual Private LAN Service";
}

identity evpn {
  base l2vpn-type;
  description
    "Ethernet VPN";
}

identity management {
  description
    "Base identity for site management scheme.";
}

identity co-managed {
  base management;
  description
    "Base identity for co-managed site.";
}

identity customer-managed {
  base management;
  description
    "Base identity for customer managed site.";
}

identity provider-managed {
  base management;
  description
    "Base identity for provider managed site.";
}

identity address-family {
  description
    "Base identity for an address family.";
}

identity ipv4 {
  base address-family;
```

```
      description
        "Identity for IPv4 address family.";
    }

    identity ipv6 {
      base address-family;
      description
        "Identity for IPv6 address family.";
    }

    identity vpn-topology {
      description
        "Base identity for VPN topology.";
    }

    identity any-to-any {
      base vpn-topology;
      description
        "Identity for any to any VPN topology.";
    }

    identity hub-spoke {
      base vpn-topology;
      description
        "Identity for Hub'n'Spoke VPN topology.";
    }

    identity hub-spoke-disjoint {
      base vpn-topology;
      description
        "Identity for Hub'n'Spoke VPN topology
         where Hubs cannot talk between each other.";

    }

    identity site-role {
      description
        "Base identity for site type.";
    }

    identity any-to-any-role {
      base site-role;
      description
        "Site in an any to any IPVPN.";
    }

    identity spoke-role {
      base site-role;
```

```
      description
        "Spoke Site in a Hub &amp;amp; Spoke IPVPN.";
    }

    identity hub-role {
      base site-role;
      description
        "Hub Site in a Hub &amp;amp; Spoke IPVPN.";
    }

    identity pm-type {
      description
        "Performance monitor type";
    }

    identity loss {
      base pm-type;
      description
        "Loss measurement";
    }

    identity delay {
      base pm-type;
      description
        "Delay measurement";
    }

    identity fault-alarm-defect-type {
      description
        "Indicating the alarm priority defect";
    }

    identity remote-rdi {
      base fault-alarm-defect-type;
      description
        "Indicates the aggregate health of the remote MEPs.";
    }

    identity remote-mac-error {
      base fault-alarm-defect-type;
      description
        "Indicates that one or more of the remote MEPs is
         reporting a failure in its Port Status TLV or
         Interface Status TLV.";
    }

    identity remote-invalid-ccm {
      base fault-alarm-defect-type;
```

```
       description
         "Indicates that at least one of the Remote MEP
          state machines is not receiving valid CCMs
          from its remote MEP.";
       }

       identity invalid-ccm {
         base fault-alarm-defect-type;
         description
           "Indicates that one or more invalid CCMs has been
            received and that 3.5 times that CCMs transmission
            interval has not yet expired.";
       }

       identity cross-connect-ccm {
         base fault-alarm-defect-type;
         description
           "Indicates that one or more cross connect CCMs has been
            received and that 3.5 times of at least one of those
            CCMs transmission interval has not yet expired.";
       }

      /* Groupings */

      grouping customer-info-grouping {
        list customer-info {
          key "customer-account-number customer-name";
          leaf customer-account-number {
            type uint32;
            description
              "Customer account number";
          }
          leaf customer-name {
            type string;
            description
              "Customer name";
          }
          container customer-operation-center {
            leaf customer-noc-street-address {
              type string;
              description
                "Customer NOC street Address.";
            }
            container customer-noc-phone-number {
              leaf main-phone-num {
                type uint32;
                description
                  "Main phone number.";
```

```
            }
            leaf extension-options {
              type uint32;
              description
                "Extension or options";
            }
            description
              "Configuration of customer NOCc phone number";
          }
          description
            "Configuration of customer operation center";
        }
        description
          "List of customer information";
      }
      description
        "Grouping for customer information";
    }

    grouping site-device {
      container device {
       list devices {
        key "device-id";
        leaf device-id {
         type string;
         description
         "Device ID";
        }
        leaf site-name {
          type string;
          description
            "Site name";
        }
        leaf address {
          type inet:ip-address;
          description
          "Address";
        }
        leaf management-transport {
          type identityref {
            base address-family;
          }
          description
            "Transport protocol used for management.";
        }
        description
          "List of devices";
       }
```

```
        description
          "Devices configuration";
      }
      description
        "Device parameters for the site.";
    }

    grouping site-management {
      container managemnt {
        leaf type {
          type identityref {
            base management;
          }
          description
            "Management type of the connection.";
        }
        description
          "Container for management";
      }
     description
       "Grouping for management";
    }

    grouping customer-location-info {
      container location {
        leaf address {
          type string;
          description
            "Address (number and street) of the site.";
        }
        leaf zip-code {
          type string;
          description
            "ZIP code of the site.";
        }
        leaf state {
          type string;
          description
            "State of the site. This leaf can also be used to
            describe a region for country who does not have
            states.";
        }
        leaf city {
          type string;
          description
            "City of the site.";
        }
        leaf country-code {
```

```
          type string;
          description
            "Country of the site.";
        }
        description
          "Location of the site.";
      }
      description
        "This grouping defines customer location parameters";
    }

    grouping site-diversity {
      container site-diversity {
        if-feature site-diversity;
        container groups {
          list group {
            key group-id;
            leaf group-id {
              type string;
              description
                "Group-id the site is belonging to";
              }
            description
              "List of group-id";
          }
          description
            "Groups the site is belonging to.
             All site network accesses will inherit those group
             values.";
        }
        description
          "Diversity constraint type.";
      }
      description
        "This grouping defines site diversity parameters";
    }

    grouping site-service {
      leaf svlan-id-ethernet-tag {
        type string;
         description
           "SVLAN-ID/Ethernet Tag configurations";
      }
      leaf cvlan-id-to-evc-map {
       type string;
        description
          "List of CVLAN-ID to EVC Map configurations";
      }
```

```
        leaf service-level-mac-limit {
          type string;
          description
            "Service-level MAC-limit (E-LAN only)";
        }
        description
          "This grouping defines site service parameters";
      }

      grouping service-protection {
        container service-protection {
          container protection-model {
            description
              "Container of protection model configurations";
          }
          container peer-evc-id {
            description
              "Container of peer EVC ID configurations";
          }
          description
            "Container of End-to-end Service Protection
            configurations";
        }
        description
          "Grouping for service protection";
      }

      grouping ethernet-service-type {
        choice ethernet-svc-type {
          case e-line {
            leaf epl {
              type boolean;
              description
                "Ethernet private line";
            }
            leaf evpl {
              type boolean;
              description
                "Ethernet virtual private line";
            }
            description
              "Case of e-line";
          }
          case e-lan {
            leaf ep-lan {
              type boolean;
              description
                "Ethernet private LAN";
```

```
            }
            leaf evp-lan {
              type boolean;
              description
                "Ethernet virtual private LAN";
            }
            description
              "Case of e-lan";
          }
          case e-access {
            leaf access-epl {
              type boolean;
              description
                "Access Ethernet virtual private line";
            }
            leaf access-evpl {
              type boolean;
              description
                "Access Ethernet virtual private line";
            }
            description
              "Case of e-access.";
          }
          description
            "Choice of Ethernet service type";
        }
        description
          "Grouping for Ethernet service type.";
      }

      grouping signaling-option-grouping {
        list signaling-option {
          key "name type";
          leaf name {
            type string;
            description
              "VRF/VFI/EVI Name";
          }
          leaf type {
            type identityref {
              base vpn-signaling-type;
            }
            description
              "VPN signaling types";
          }
          container mp-bgp-l2vpn {
            leaf vpn-id {
              type string;
```

```
            description
              "Identifies the target VPN";
          }
          leaf type {
            type identityref {
              base l2vpn-type;
            }
            description
              "L2VPN types";
          }
          description
            "Container for MP BGP L2VPN";
        }
        container mp-bgp-evpn {
          leaf vpn-id {
            type string;
            description
              "Identifies the target VPN";
          }
          leaf type {
            type identityref {
              base l2vpn-type;
            }
            description
              "L2VPN types";
          }
          description
            "Container for MP BGP L2VPN";
        }
        container t-ldp-pwe {
          list PE-EG-list {
            key "service-ip-lo-addr vc-id";
            leaf service-ip-lo-addr {
              type inet:ip-address;
              description
                "Service ip lo address";
            }
            leaf vc-id {
              type string;
              description
              "VC id";
            }
            description
              "List of PE/EG";
          }
          description
            "Container of T-LDP PWE configurations";
        }
```

```
          container pwe-encapsulation-type {
            leaf ethernet {
              type boolean;
              description
                "Ethernet";
            }
            leaf vlan {
              type boolean;
              description
                "VLAN";
            }
            description
              "Container of PWE Encapsulation Type configurations";
          }
          container pwe-mtu {
            leaf allow-mtu-mismatch {
              type boolean;
              description
                "Allow MTU mismatch";
            }
            description
              "Container of PWE MTU configurations";
          }
          container control-word {
            description
              "Container of control word configurations";
          }
          description
            "List of VPN Signaling Option.";
        }
        description
          "Grouping for signaling option";
      }

      grouping load-balance-grouping {
        leaf fat-pw {
          type boolean;
          description
            "Fat label is applied to Pseudowires across MPLS
             network";
        }
        leaf entropy-label {
          type boolean;
          description
            "Entropy label is applied to IP forwarding,
             L2VPN or L3VPN across MPLS network";
        }
        leaf vxlan-source-port {
```

```
          type string;
          description
            "Vxlan source port";
        }
      description
        "Grouping for load balance ";
    }

    grouping intra-mkt-grouping {
      list intra-mkt {
        key "metro-mkt-id mkt-name";
        leaf metro-mkt-id {
          type uint32;
          description
            "Metro MKT ID";
        }
        leaf mkt-name {
          type string;
          description
            "MKT Name";
        }
        description
          "List of intra-MKT";
      }
      description
        "Grouping for intra-MKT";
    }

    grouping inter-mkt-service {
      leaf inter-mkt-service{
        type boolean;
        description
          "Indicate whether service is inter market service.";
      }
      description
        "Grouping for inter-MKT service";
    }

    grouping evc-id-grouping {
      leaf evc-id {
        type boolean;
        description
          "Ethernet Virtual Connection identifier";
      }
      description
        "Grouping for EVC-ID";
    }
```

```
     grouping svc-type-grouping {
       container svc-type {
         container evc {
           leaf evc-id {
             type boolean;
             description
               "Indicate whether the Ethernet virtual connection
                id support.";
           }
           leaf number-of-pe {
             type uint32;
             config false;
             description
               "Number of PEs";
           }
           leaf number-of-site {
             type uint32;
             config false;
             description
               "Number of Sites";
           }
           description
             "Container for Ethernet virtual connection.";
         }
         container ovc {
           leaf on-net-ovc-id {
             type boolean;
             description
               "Indicate whether the on net OVC id support.";
           }
           leaf off-net-ov-id {
             type boolean;
             description
               "Indicate whether the off net OVC id support.";
           }
           description
             "Container for OVC";
         }
         description
           "Container for service types.";
       }
       description
         "Grouping of service types.";
     }

     grouping cfm-802-grouping {
       leaf MAID {
         type string;
```

```
        description
          "MA ID";
      }
      leaf mep-id {
       type uint32;
       description
         "Local MEP ID";
      }
      leaf mep-level {
        type uint32;
        description
          "MEP level";
      }
      leaf mep-up-down {
        type enumeration {
          enum up {
            description
              "MEP up";
          }
          enum down {
            description
              "MEP down";
          }
        }
        description
          "MEP up/down";
      }
      leaf remote-mep-id {
        type uint32;
        description
          "Remote MEP ID";
      }
      leaf cos-for-cfm-pdus {
        type uint32;
        description
          "COS for CFM PDUs";
      }
      leaf ccm-interval {
        type uint32;
        description
          "CCM interval";
      }
      leaf ccm-holdtime {
        type uint32;
        description
          "CCM hold time";
      }
      leaf alarm-priority-defect {
```

```
        type identityref {
          base fault-alarm-defect-type;
        }
        description
         "The lowest priority defect that is
          allowed to generate a Fault Alarm.
          The non-existence of this leaf means
          that no defects are to be reported";
      }
      leaf ccm-p-bits-pri {
        type ccm-priority-type;
        description
          "The priority parameter for CCMs transmitted by the MEP";
      }
      description
        "Grouping for 802.1ag CFM attribute";
    }

    grouping y-1731{
      list y-1731 {
        key MAID;
        leaf MAID {
          type string;
          description
            "MA ID ";
        }
        leaf mep-id {
          type uint32;
          description
            "Local MEP ID";
        }
        leaf type {
          type identityref {
            base pm-type;
          }
          description
            "Performance monitor types";
        }
        leaf remote-mep-id {
          type uint32;
          description
            "Remote MEP ID";
        }
        leaf message-period {
          type uint32;
          description
            "Defines the interval between OAM messages. The message
             period is expressed in milliseconds";
```

```
        }
        leaf measurement-interval {
          type uint32;
          description
            "Specifies the measurement interval for statistics. The
             measurement interval is expressed in seconds";
        }
        leaf cos {
          type uint32;
          description
            "Class of service";
        }
        leaf loss-measurement {
          type boolean;
          description
            "Whether enable loss measurement";
        }
        leaf synthethic-loss-measurement {
          type boolean;
          description
            "Indicate whether enable synthetic loss measurement";
        }
        container delay-measurement {
          leaf enable-dm {
            type boolean;
            description
              "Whether to enable delay measurement";
          }
          leaf two-way {
            type boolean;
            description
              "Whether delay measurement is two-way (true) of one-
              way (false)";
          }
          description
            "Container for delay measurement";
        }
        leaf frame-size {
          type uint32;
          description
            "Frame size";
        }
        leaf session-type {
          type enumeration {
            enum proactive {
              description
                "Proactive mode";
            }
```

```
            enum on-demand {
              description
                "On demand mode";
            }
          }
          description
            "Session type";
        }
        description
          "List for y-1731.";
      }
      description
        "Grouping for y.1731";
    }

    grouping enni-site-info-grouping {
      container site-info {
        leaf site-name {
          type string;
          description
            "Site name";
        }
        leaf address {
          type inet:ip-address;
          description
            "Address";
        }
        leaf Edge-Gateway-Device-Info {
          type string;
          description
            "Edge Gateway Device Info ";
        }
        description
          "Container of site info configurations";
      }
      description
        "Grouping for site information";
    }

    grouping site-security {
      container security {
        description
          "Security parameters";
      }
      description
        "This grouping defines security parameters for a site";
    }
```

```
     grouping lacp-grouping {
       container LACP {
         leaf LACP-state {
           type identityref {
             base lacp-state;
           }
           description
             "LACP on/off";
         }
         leaf LACP-mode {
           type identityref {
             base lacp-mode;
           }
           description
             "LACP mode";
         }
         leaf LACP-speed {
           type identityref {
             base lacp-speed;
           }
           description
             "LACP speed";
         }
         leaf mini-link {
           type uint32;
           description
             "Mini link";
         }
         leaf system-priority {
           type uint16;
           description
             "Indicates the LACP priority for the system.
              The range is from 0 to 65535.
              The default is 32768.";
         }
         container Micro-BFD {
           if-feature Micro-BFD;
           leaf Micro-BFD-on-off {
             type enumeration {
               enum on {
                 description
                   "Micro-bfd on";
               }
               enum off {
                 description
                   "Micro-bfd off";
               }
             }
```

```
            description
              "Micro BFD ON/OFF";
          }
          leaf bfd-interval {
            type uint32;
            description
              "BFD interval";
          }
          leaf bfd-hold-timer {
            type uint32;
            description
              "BFD hold timer";
          }
          description
            "Container of Micro-BFD configurations";
        }
        container bfd {
          if-feature bfd;
          leaf bfd-enabled {
            type boolean;
            description
              "BFD activation";
          }
          choice holdtime {
            case profile {
              leaf profile-name {
                type string;
                description
                  "Service provider well known profile.";
              }
              description
                "Service provider well known profile.";
            }
            case fixed {
              leaf fixed-value {
                type uint32;
                units msec;
                description
                  "Expected hold time expressed in msec.";
              }
            }
            description
              "Choice for hold time flavor.";
          }
          description
            "Container for BFD.";
        }
        container Member-link-list {
```

```
            list member-link {
              key "name";
              leaf name {
                type string;
                description
                  "Member link name";
              }
              leaf port-speed {
                type uint32;
                description
                  "Port speed";
              }
              leaf auto-neg {
                type string;
                description
                  "Auto neg";
              }
              leaf mtu {
                type uint32;
                description
                  "MTU";
              }
              container oam-802.3AH-link {
                if-feature oam-3ah;
                leaf enable {
                  type boolean;
                  description
                    "Indicate whether support oam 802.3 ah link";
                }
                description
                  "Container for oam 802.3 ah link.";
              }
              description
                "Member link";
            }
            description
              "Container of Member link list";
          }
          leaf flow-control {
            type string;
            description
              "Flow control";
          }
          leaf encapsulation-type {
            type enumeration {
              enum VLAN {
                description
                  "VLAN";
```

```
            }
            enum ether {
              description
                "Ethernet";
            }
          }
          description
            "Encapsulation type";
        }
        leaf ethertype {
          type string;
          description
            "Ether type";
        }
        leaf lldp {
          type boolean;
           description
             "LLDP";
        }
        description
          "LACP";
      }
      description
        "Grouping for lacp";
    }

    grouping phy-interface-grouping {
      container phy-interface {
        leaf port-number {
          type uint32;
          description
            "Port number";
        }
        leaf port-speed {
          type uint32;
          description
            "Port speed";
        }
        leaf auto-neg {
          type string;
          description
            "Auto neg";
        }
        leaf phy-mtu {
          type uint32;
          description
            "PHY MTU";
        }
```

```
        leaf flow-control {
          type string;
          description
            "Flow control";
        }
        leaf encapsulation-type {
          type enumeration {
            enum VLAN {
              description
                "VLAN";
            }
            enum Ethernet {
              description
                "Ethernet";
            }
          }
          description
            "Encapsulation-type";
        }
        leaf ethertype {
          type string;
          description
            "Ethertype";
        }
        leaf lldp {
          type boolean;
          description
            "LLDP";
        }
        container oam-802.3AH-link{
         if-feature oam-3ah;
         leaf enable {
           type boolean;
           description
             "Indicate whether support oam 802.3 ah link";
         }
         description
           "Container for oam 802.3 ah link.";
        }
        leaf uni-loop-prevention {
          type boolean;
          description
            "If this leaf set to truth that the port automatically
             goes down when a physical loopback is detect.";
        }
        description
          "Container of PHY Interface Attributes configurations";
      }
```

```
      description
        "Grouping for phy interface.";
    }

    grouping lag-interface-grouping {
      container LAG-interface {
        list LAG-interface {
          key "LAG-interface-number";
          leaf LAG-interface-number {
            type uint32;
            description
              "LAG interface number";
          }
          uses lacp-grouping;
          description
            "List of LAG interfaces";
        }
        description
          "Container of LAG interface attributes configuration";
      }
      description
        "Grouping for LAG interface";
    }

    grouping bearer-grouping {
      container bearer {
        uses phy-interface-grouping;
        uses lag-interface-grouping;
        leaf interface-description {
          type string;
          description
            "Interface description";
        }
        leaf sub-if-id {
          type uint32;
          description
            "Sub-if id";
        }
        description
          "Container for bearer";
      }
      description
      "Grouping for bearer.";
    }

    grouping ethernet-connection-grouping {
      container ethernet-connection {
        container vlan {
```

```
          leaf svlan-id-ethernet-tag {
            type string;
            description
              "SVLAN-ID/Ethernet Tag configurations";
          }
          description
            "Abstract container for VLAN";
        }
        description
          "Container for Ethernet connection";
      }
      description
        "Grouping for Ethernet connection";
    }

    grouping evc-mtu-grouping {
      leaf evc-mtu {
        type uint32;
        description
          "EVC MTU";
      }
      description
        "Grouping for evc mtu";
    }

    grouping mac-addr-limit-grouping {
      container mac-addr-limit {
        leaf exceeding-option {
          type uint32;
          description
            "Exceeding option";
        }
        description
          "Container of MAC-Addr limit configurations";
      }
      description
        "Grouping for mac address limit";
    }

    grouping multihoming-grouping {
      container multihoming {
        list multihoming {
          key "ESI";
          leaf ESI {
            type string;
            description
              "Ethernet segment id";
          }
```

```
          choice redundancy-mode {
            case single-active {
              leaf single-active {
                type boolean;
                description
                  "Single active";
              }
              description
                "Single active case";
            }
            case all-active {
              leaf all-active {
                type boolean;
                description
                  "All active";
              }
              description
                "All active case";
            }
            description
              "Redundancy mode choice";
          }
          description
            "List of multihomings";
        }
        description
          "Container of multihoming optional configurations";
      }
      description
        "Grouping for multihoming";
    }

    grouping l2cp-grouping {
      container L2CP-control {
        leaf stp-rstp-mstp {
          type control-mode;
          description
            "STP/RSTP/MSTP";
        }
        leaf pause {
          type control-mode;
          description
            "Pause";
        }
        leaf lacp-lamp {
          type control-mode;
          description
            "LACP/LAMP";
```

```
        }
        leaf link-oam {
          type control-mode;
          description
            "Link OAM";
        }
        leaf esmc {
          type control-mode;
          description
            "ESMC";
        }
        leaf l2cp-802.1x {
          type control-mode;
          description
            "802.x";
        }
        leaf e-lmi {
          type control-mode;
          description
            "E-LMI";
        }
        leaf lldp {
          type boolean;
          description
            "LLDP";
        }
        leaf ptp-peer-delay {
          type control-mode;
          description
            "PTP peer delay";
        }
        leaf garp-mrp {
          type control-mode;
          description
            "GARP/MARP";
        }
        leaf provider-bridge-group {
          type yang:mac-address;
          description
            "Provider bridge group reserved MAC address
             01-80-C2-00-00-08";
        }
        leaf provider-bridge-mvrp {
          type yang:mac-address;
          description
            "Provider bridge MVRP reserved MAC address
             01-80-C2-00-00-0D";
        }
```

```
         description
           "Container of L2CP control configurations";
        }
        description
          "Grouping for l2cp control";
     }

     grouping service-level-grouping {
       container service-level {
         leaf cos-identifier {
           type identityref {
            base cos-id;
           }
           description
             "COS Identifier [ EVC | EVC + PCP ]";
         }
         leaf color-identifier {
           type identityref {
            base color-id;
           }
           description
             "Color Identifier [ EVC | EVC + CVLAN ]";
         }
         leaf ingress-bw-profile-per-evc {
           type string;
           description
             "Ingress Bandwidth Profile per EVC";
         }
         leaf ingress-bw-profile-per-cos-id {
           type string;
           description
             "Ingress Bandwidth Profile per COS Identifier";
         }
         leaf egress-bw-profile-per-evc {
           type string;
           description
             "Egress Bandwidth Profile per EVC";
         }
         leaf egress-bw-profile-per-cos-id {
           type string;
           description
             "Egress Bandwidth Profile per COS Identifier";
         }
         leaf byte-offset {
           type uint16;
           description
             "For not including extra VLAN tags in the QoS
             calculation";
```

```
        }
        leaf policing {
         type identityref {
          base policing;
         }
         description
         "The policing can be either one-rate,
         two-color (1R2C) or two-rate, three-color (2R3C)";
        }
        leaf performance-tier-option {
          type identityref {
            base performance-tier-option;
          }
          description
            "Performance tier option";
        }
        leaf COS {
          type uint32;
          description
            "Class of Service";
        }
        description
          "Container of service level configurations.";
      }
      description
        "Grouping for service level.";
    }

    grouping B-U-M-strom-control-grouping {
      container B-U-M-strom-control {
         leaf BUM-overall-rate {
           type uint32;
           description
             "overall rate for BUM";
          }
          list BUM-rate-per-type {
            key "type";
            leaf type {
              type identityref {
                base BUM-type;
              }
              description
                "BUM type";
            }
            leaf rate {
              type uint32;
              description
                "rate for BUM";
```

```
          }
          description
            "List of rate per type";
        }
        description
          "Container of B-U-M-strom-control configurations";
      }
      description
        "Grouping for B-U-M-strom-control";
    }

    grouping mac-loop-prevention-grouping {
      container mac-loop-prevention {
        leaf frequency {
          type uint32;
          description
            "Frequency";
        }
        leaf protection-type {
          type identityref {
            base loop-prevention-type;
          }
          description
            "Protection type";
        }
        leaf number-retries {
          type uint32;
          description
            "Number of retries";
        }
        description
          "Container of MAC loop prevention.";
      }
      description
        "Grouping for MAC loop prevention";
    }

    grouping ethernet-svc-oam-grouping {
      container Ethernet-Service-OAM {
        leaf MD-name{
          type string;
          description
            "Maintenance domain name";
        }
        leaf MD-level {
          type uint8;
          description
            "Maintenance domain level";
```

```
        }
        container cfm-802.1-ag {
          list n2-uni-c {
            key "MAID";
            uses cfm-802-grouping;
            description
              "List of UNI-N to UNI-C";
          }
          list n2-uni-n {
            key "MAID";
            uses cfm-802-grouping;
            description
             "List of UNI-N to UNI-N";
          }
          description
            "Container of 802.1ag CFM configurations.";
        }
        uses y-1731;
        description
          "Container for Ethernet service OAM.";
      }
      description
        "Grouping for Ethernet service OAM.";
    }

    grouping fate-sharing-group {
      container groups {
        leaf fate-sharing-group-size {
          type uint16;
          description
            "fate sharing group size.";
        }
        list group {
          key group-id;
          leaf group-id {
            type string;
            description
              "Group-id the site network access
               is belonging to";
          }
          description
            "List of group-id";
        }
        description
          "Groups the fate sharing group member
           is belonging to";
      }
      description
```

```
        "Grouping for Fate sharing group.";
    }

    /* MAIN L2VPN SERVICE */

    container l2vpn-svc {

      /* CUSTOMER */
      container customer-info {
        uses customer-info-grouping;
        description
          "Container of customer information configurations.";
      }

      /* SERVICE */
      container vpn-services {
        list vpn-svc {
          key "svc-id";
          leaf svc-id {
            type string;
            description
              "Defining a service id.";
          }
          uses svc-type-grouping;
          container ethernet-svc-type {
            uses ethernet-service-type;
            description
              "Container of Ethernet service type";
          }
          container metro-network-id {
            uses inter-mkt-service;
            uses intra-mkt-grouping;
            description
              "Container of Metro-Network ID configurations";
          }
          container signaling-option {
            uses signaling-option-grouping;
            description
              "Container for signaling option";
          }
          container load-balance-options{
            uses load-balance-grouping;
            description
              "Container for load balance options";
          }
          uses site-service;
          uses service-protection;
          container sla-targets {
```

```
            description
              "Container for SLA targets";
          }
          description
            "List of vpn-svc";
        }
        description
          "Container of vpn-services configurations";
      }

      /* SITE */
      container sites {
        list site {
          key "site-id site-type";
          leaf site-id {
            type string;
            description
              "Site id";
          }
          leaf site-type {
            type identityref {
              base site-type;
            }
            description
              "Site type";
          }
          uses site-device;
          uses site-management;
          uses customer-location-info;
          uses site-diversity;
          uses site-security;
          container signaling-option {
            if-feature signaling-option;
            uses signaling-option-grouping;
            description
              "Container for signaling option";
          }
          container load-balance-options {
            uses load-balance-grouping;
            description
              "Container for load balance options";
          }
          container ports {
            list port {
              key "id";
              leaf id {
                type string;
                description
```

```
                "Identifier";
            }
            leaf remote-carrier-name {
              when "'../site-type' = 'enni'" {
                description
                  "Site type = enni";
              }
              type string;
              description
                "Remote carrier name";
            }
            uses fate-sharing-group;
            uses bearer-grouping;
            uses ethernet-connection-grouping;
            uses evc-mtu-grouping;
            uses mac-addr-limit-grouping;
            uses multihoming-grouping;
            uses l2cp-grouping;
            container service {
              uses site-service;
              uses service-level-grouping;
              description
                "Container for site service.";
            }
            uses B-U-M-strom-control-grouping;
            uses mac-loop-prevention-grouping;
            uses ethernet-svc-oam-grouping;
            uses site-security;
            description
              "List of ports";
          }
          description
            "Container of port configurations";
        }
        description
          "List of sites";
      }
      description
        "Container of site configurations";
    }
    description
      "Container of l2vpn-svc configurations";
  }
}
<CODE ENDS>
```

9. Security Considerations

The YANG modules defined in this document MAY be accessed via the RESTCONF protocol [I-D.ietf-netconf-restconf] or NETCONF protocol ([RFC6241]. The lowest RESTCONF or NETCONF layer requires that the transport-layer protocol provides both data integrity and confidentiality, see Section 2 in [I-D.ietf-netconf-restconf] and [RFC6241]. The client MUST carefully examine the certificate presented by the server to determine if it meets the client's expectations, and the server MUST authenticate client access to any protected resource. The client identity derived from the authentication mechanism used is subject to the NETCONF Access Control Module (NACM) ([RFC6536]). Other protocols to access this YANG module are also required to support the similar mechanism.

The data nodes defined in the "ietf-l2vpn-svc" YANG module MUST be carefully created/read/updated/deleted. The entries in the lists below include customer proprietary or confidential information, therefore only authorized clients MUST access the information and the other clients MUST NOT be able to access to the information.

- /l2vpn-svc/vpn-services/vpn-svc

- /l2vpn-svc/sites/site

10. Acknowledgements

Thanks to Qin Wu and Adrian Farrel for facilitating work on the initial revisions of this document.

This document has drawn on the work of the L3SM Working Group expressed in [I-D.ietf-l3sm-l3vpn-service-model].

11. IANA Considerations

IANA is requested to assign a new URI from the IETF XML registry ([RFC3688]). The following URI is suggested:

```
URI: urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc
Registrant Contact: L2SM WG
XML: N/A, the requested URI is an XML namespace
```

This document also requests a new YANG module name in the YANG Module Names registry ([RFC6020]) with the following suggestion:

```
name: ietf-l2vpn-svc
namespace: urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc
prefix: l2vpn-svc
reference: RFC XXXX
```

12. References

12.1. Normative References

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-18 (work in progress), October 2016.

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

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[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>.

[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>.

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

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

[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>.

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

12.2. Informative References

[I-D.ietf-bess-evpn-yang]
Brissette, P., Sajassi, A., Shah, H., Li, Z., Tiruveedhula, K., Hussain, I., and J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-01 (work in progress), July 2016.

[I-D.ietf-bess-l2vpn-yang]
Shah, H., Brissette, P., Chen, I., Hussain, I., and B. Wen, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-00 (work in progress), June 2016.

[I-D.ietf-l3sm-l3vpn-service-model]
Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN service delivery", draft-ietf-l3sm-l3vpn-service-model-18 (work in progress), October 2016.

[I-D.wu-opsawg-service-model-explained]
Wu, Q., LIU, S., and A. Farrel, "Service Models Explained", draft-wu-opsawg-service-model-explained-03 (work in progress), September 2016.

[IEEE-802-1ag]
IEEE, "802.1ag - Connectivity Fault Management", December 2007.

[ITU-T-Y-1731]
ITU-T, "Recommendation Y.1731 - OAM functions and mechanisms for Ethernet based networks", February 2008.

[MEF-23-2]
MEF Forum, "Implementation Agreement MEF 23.2 : Carrier Ethernet Class of Service - Phase 3", August 2016.

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

Authors' Addresses

Bin Wen
Comcast

Email: Bin_Wen@comcast.com

Giuseppe Fioccola
Telecom Italia

Email: giuseppe.fioccola@telecomitalia.it

Chongfeng Xie
China Telecom

Email: xiechf@ctbri.com.cn

Luay Jalil
Verizon

Email: luay.jalil@verizon.com