L3SM Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2016

C. Xie
A. Wang
China Telecom
R. Schott
Deutsche Telekom
D. Zhang
November 1, 2015

YANG Data Model for L2VPN service
draft-xie-l3sm-l2vpn-service-model-01

Abstract

This document provides an example of service yang data model for layer 2 provider provisioned VPN service. Unlike L3VPN, L2VPN doesn't provide L3 interface to customer using IP infrastructure or doesn't provide IP connectivity between pairs of customer sites. Therefore straight augment the l3vpn model with l2vpn parameters may not be appropriate. However [draft-ietf-l3sm-l3vpn-service-model] has defined a lot of reusable groupings such as operational-requirements, customer-location-info, site-diversity ,site-availability,etc. In this document, we reuse these common groupings and add some l2vpn parameters to develop the l2vpn service model.

Similar to the l3vpn service model, this model provides an abstracted view of the Layer 2 service configuration components. It will be 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.

Table of Contents

1. Introduction

Layer 2 VPN emulates the behavior of a local area network (LAN) across an internet protocol (IP) or MPLS-enabled IP network allowing Ethernet devices to communicate with each other as if they were connected to a common LAN segment[RFC4664]. Building a L2VPN system requires coordination between the Service Provider and the customer. The Service Provider provides L2 connectivity; the customer builds a network using data link resources obtained from the Service Provider. In an L2VPN service, the Service Provider does not require information about a customer's network topology, policies, routing information, point-to-point links.

The Service Provider only requires Provider Edge (PE) routers with the following capabilities:

- Encapsulation of L2 protocol data units (PDU) into layer 3 packets
- Inter-connection of any-to-any L2 transports.
- Emulation of L2 quality-of-service (QoS) over a packet switch network.
- Ease of configuration of the L2 service.
- Support for different types of tunneling mechanisms (MPLS, L2TPv3, IPSec, GRE, and others)
- L2VPN process databases include all information related to circuits and their connections.

This document provides an example of service model for Layer 2 VPN service. It can be used by a management system as an input then choice suited configurations models to configure the different network elements to deliver the service.

2. Conventions and Terminology

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

The following notations are used within the data tree and carry the meaning as below.

```
  Each node is printed as:
    <status> <flags> <name> <opts> <type>

    <status> is one of:
         +  for current
         x  for deprecated
         o  for obsolete
    <flags> is one of:
        rw for configuration data
        ro for non-configuration data
        -x for rpcs
        -n for notifications
    <name> is the name of the node

    If the node is augmented into the tree from another module, its name
    is printed as <prefix>:<name>.
    <opts> is one of:
         ?  for an optional leaf or choice
         !  for a presence container
         *  for a leaf-list or list
         [<keys>] for a list's keys
    <type> is the name of the type for leafs and leaf-lists
```

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. Terminologies

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.

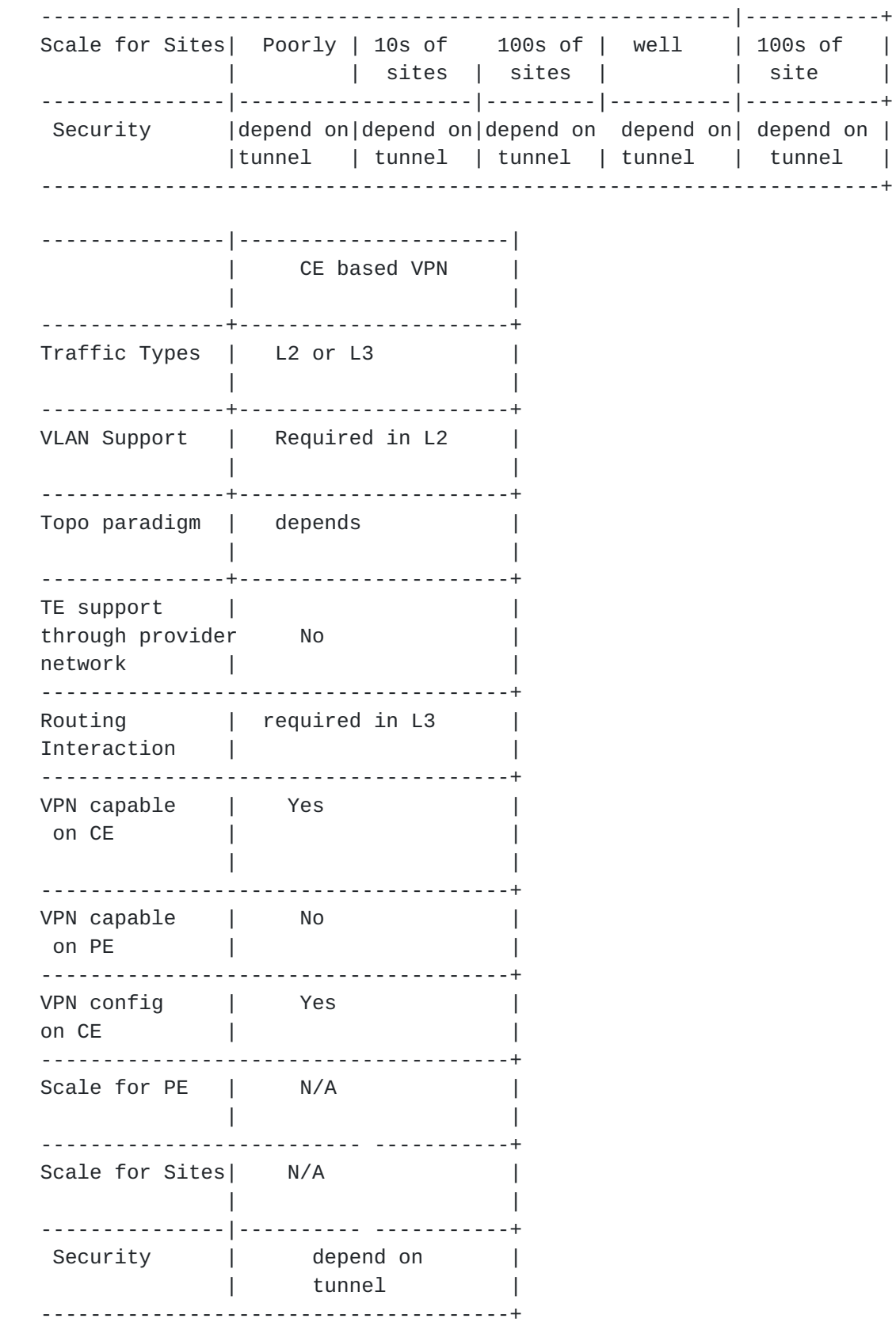
VPW A Virtual Private Wire Service (VPWS) is a point-to-point circuit (link) connecting two Customer Edge devices. The link is established as a logical through a packet switched network. The CE in the customer network is connected to a PE in the provider network via an Attachment Circuit; the Attachment Circuit is either a physical or a logical circuit.

3. L2VPN and L3VPN comparison

There are two fundamentally different kinds of Layer 2 VPN service that a service provider could offer to a customer: Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS). There is also the possibility of an IP-only LAN-like Service (IPLS)[RFC4664]. The VPN service must match the type of service required by the VPN user. Different VPN solutions offer either layer 2 or layer 3 connectivity between VPN sites.

Below is a table for comparison analysis between L2VPN and L3VPN service.

	PE-based Layer 2			PE-based Layer 3	
	VPWs	VPLS	IPLS	RFC4364	vRouter
Traffic Types	ATM/FR	Ethernet	IP over Ethernet	IPv4 and IPv6	
VLAN support	Depends	Yes	Depends	No	
Topology paradigm	p2p	any to any hub spoke	Depends	any to any, hub spoke disjoint	
TE support through provider network	Yes	Yes	Yes	Yes	Yes
Routing Interaction	No	No	No	Yes	
VPN capable on CE	No	No	No	No	No
VPN capable on PE	Yes	Yes	Yes	Yes	Yes
VPN config on CE	some	No	No	No	No
Scale for PE	well	not well unless distributed	well	well	not well


Scale for Sites
Poorly
10s of sites
100s of sites
well
100s of site
Security
depend on tunnel
depend on tunnel
depend on tunnel
depend on tunnel
depend on tunnel
CE based VPN
Traffic Types
L2 or L3
VLAN Support
Required in L2
Topo paradigm
depends
TE support through provider network
No
Routing Interaction
required in L3
VPN capable on CE
Yes
VPN capable on PE
No
VPN config on CE
Yes
Scale for PE
N/A
Scale for Sites
N/A
Security
depend on tunnel

Compared with L3VPN, L2VPN has High scalability sinceMPLS L2VPN establishes only Layer 2 connections and It does not involve the routing information of users. This greatly reduces the load of the PEs and even the load of the whole service provider network, enabling carriers to support more VPNs and more users. In addition, L2VPN provides Guaranteed reliability and private routing information security since no routing information of users is involved, L2VPN neither tries to obtain nor processes the routing information of users, guaranteeing the security of the user VPN routing information.

4. Layer 2 VPN service model design

4.1. Reuse the groupings defined in L3SM service model

[RFC3809] provides requirements that are generic to both Layer 2 Virtual Private Networks (L2VPN) and Layer 3 Virtual Private Networks (L3VPN).These requirements are independent of any particular type of PPVPN technology and include service, provider and engineering requirements. In this document, we reuse some common groupings corresponding to these requirements which are defined in the [L3VPN-svc].

The following table summarizes the common grouping which are used in this document:

grouping name:

vpn-svc-cfg
operational-requirements
customer-location-info
site-diversity
site-availability
site-management
site-vpn-policy
site-security-authentication
site-security-encryption
site-security-acl
site-service-protection
site-service-mpls
site-service-multicast

4.2. Customer lan connection

In this document, we analyzed the different of L2VPN and L3VPN in section 2. The major differences are traffic type and connectivity type, e.g., Layer 2 services is usually based on frame relay and asynchronous transfer mode (ATM) while Layer 3 service is based on IPv4 and IPv6.

In Layer 2 VPN, The VC labels are used by the PE routers for demultiplexing traffic arriving from different L2VPN services over the same set of tunnel/PW. And the MAC address also be used in the layer 2 customer lan connection:

```
|  +--rw customer-specific-information
|  |  ......
|  |  +--rw customer-lan-connection* [address]
|  |  |  +--rw address         union
|  |  |  +--rw lan-protocol?   identityref
|  |  |  +--rw vc-label?       string
|  |  |  +--rw mac-address?    yang:mac-address
```

4.3. Attachment

In layer 2 VPN, the physical parameters of the attachment may be a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, a PPP connection on a physical interface, etc. To make it easy to be extended, in this document we define a bearer identity and several other identities which are base on the bearer identity:

```
identity bearer{
 description
     "base identity of bearer.";
}

identity ems{
 base bearer;
     description
     "identity of vpls ethernet mulitpoint service.";
}

identity emrs{
 base bearer;
 description
 "identity of vpls ethernet multipoint relay service.";
}

identity fr{
 base bearer;
     description
     "identity of Frame Relay";
}

identity ethernet{
 base bearer;
     description
     "identity of ethernet.";
}

identity atm{
 base bearer;
     description
     "identity of ATM.";
}

identity ppp-or-hdlc{
 base bearer;
     description
     "identity of PPP/HDLC.";
}
```

4.4. QoS

For QoS service, the match-flow of L2VPN are quite different from L3VPN's. The source/destination MAC address, local/remote label, vlan id may be used:

```
        +--rw service
           ......
           +--rw qos
           |  +--rw qos-classification-policy
           |     +--rw rules* [id]
           |        +--rw id                    uint16
           |        +--rw match-flow
           |        |  +--rw dest-mac-address?   yang:mac-address
           |        |  +--rw src-mac-address?    yang:mac-address
           |        |  +--rw local-label?        string
           |        |  +--rw remote-label?       string
           |        |  +--rw dot1q-vlan-bitmap   string
           |        |  +--rw qinq-svlan-bitmap   string
           |        |  +--rw qinq-cvlan-bitmap   string
           |        |  +--rw target-class-id?    string
           |        +--rw std-qos-profile?     string
           ......
```

5. YANG Module

```
<CODE BEGINS> file "ietf-l2vpn-svc.yang"
    module ietf-l2vpn-svc {
      namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn-svc";

      prefix l2vpn-svc;

      import ietf-routing {
              prefix "rt";
          }

      import ietf-inet-types {
          prefix inet;
      }

      import ietf-yang-types {
          prefix yang;
      }

          import ietf-l3vpn-svc{
              prefix l3vpn-svc;
          }

      organization
       "IETF L3SM Working Group";

      contact
          "WG List:   <mailto:l3sm@ietf.org>
```

```
            Editor:

            ";

        description
            "The YANG module defines a generic service configuration
            model for Layer 2 VPN common across all of the vendor
            implementations.";

        revision 2015-10-12 {
            description
             "l2vpn first version.";
            reference "";
        }


    /*identity*/
      identity bearer{
       description
           "base identity of bearer.";
      }

      identity ems{
       base bearer;
           description
           "identity of vpls ethernet mulitpoint service.";
      }

      identity emrs{
       base bearer;
       description
       "identity of vpls ethernet multipoint relay service.";
      }

      identity fr{
       base bearer;
           description
           "identity of Frame Relay";
      }

      identity ethernet{
       base bearer;
           description
           "identity of ethernet.";
      }

      identity atm{
       base bearer;
```

```
          description
          "identity of ATM.";
      }

      identity ppp-or-hdlc{
       base bearer;
          description
          "identity of PPP/HDLC.";
      }
         /* Groupings */


      container l2vpn-svc{
       description
          "this container contains several "
          +"l2vpn service parameters";
       list vpn-svc{
            key "name";
            description
            "list of layer 2 vpn service";
            uses l3vpn-svc:vpn-svc-cfg;
          }

          list sites{
            key "site-id";
            description
            "list of layer 2 vpn sites";
            leaf site-id{
             type string;
             description
             "site identifier";
            }

            // apply-template

            uses l3vpn-svc:operational-requirements;
            uses l3vpn-svc:customer-location-info;
            uses l3vpn-svc:site-diversity;
            uses l3vpn-svc:site-availability;
            uses l3vpn-svc:site-management;
            uses l3vpn-svc:site-vpn-policy;
             container customer-specific-information {
                 leaf name {
                     type string;
                     description
                      "Name of the customer router.";
                 }
                 leaf autonomous-system {
```

```
                    type uint32;
                    description
                     "AS number.";
                }
                leaf interface {
                    type string;
                    description
                     "Interface reference of the access.";
                }
                list customer-lan-connection {
                    key "address";
                    leaf address {
                        type union {
                            type inet:ipv4-address;
                            type inet:ipv6-address;
                        }
                        description
                         "Address given by the customer on its LAN
                        for the SP router.";
                    }
                    leaf lan-protocol {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Transport protocol used on LAN.";
                    }
                                  leaf vc-label{
                                   type string;
                                   description
                                   "the vc label of l2vpn";
                                  }
                                  leaf mac-address{
                                   type yang:mac-address;
                                description
                                "mac address";
                                  }
                    description
                     "List of customer LAN to be connected
                     directly on the CE.";
                }
                container cascaded-lan-prefixes {
                    list ipv4-lan-prefixes {
                        key lan;

                        leaf lan {
                            type inet:ipv4-prefix;
                            description
```

```
                        "Lan prefixes.";
                    }
                    leaf lan-tag {
                        type string;
                        description
                         "Internal tag to be used in vpn
                         policies.";
                    }
                    leaf next-hop {
                        type inet:ipv4-address;
                        description
                         "Nexthop address to use at customer
                         side.";
                    }
                    description "
                        List of LAN prefixes for
                        the site.
                        ";
                }
                list ipv6-lan-prefixes {
                    key lan;

                    leaf lan {
                        type inet:ipv6-prefix;
                        description
                         "Lan prefixes.";
                    }
                    leaf lan-tag {
                        type string;
                        description
                         "Internal tag to be used
                         in vpn policies.";
                    }
                    leaf next-hop {
                        type inet:ipv6-address;
                        description
                         "Nexthop address to use at
                         customer side.";
                    }
                    description "
                        List of LAN prefixes for the site.
                        ";
                }
                description
                    "LAN prefixes from the customer.";
            }

            description
```

```
                "Customer premise configuration.";
             }

             container security{
              description
              "layer 2 vpn security parameters.";

              uses l3vpn-svc:site-security-authentication;
              uses l3vpn-svc:site-security-encryption;
              uses l3vpn-svc:site-security-acl;
             }

             container attachment{
              description
              "TBD";
              container bearer {
            leaf type {
             type identityref{
                      base bearer;
                     }
             description
              "Type of bearer Ethernet ...
              Operator specific.";
             }
            leaf bearer-reference {
              type string;
              description
              "This is an internal reference for the
               service provider.";
             }
            description
            "Bearer specific parameters.
             To be augmented.";
           }
               container l2-connection{
                leaf peer-id{
             type inet:ip-address;
                 description
                     "peer ip address.";}
                      container ipv4{
               leaf address-allocation-type {
                type identityref {
                 base l3vpn-svc:address-allocation-type;
                }
                 description
                 "Defines how addresses are allocated.
                  Need to be detailed further.";
                }
```

```
             leaf subnet-prefix {
              type inet:ipv4-prefix;
               description
               "Interco subnet.";
              }
              description
              "IPv4 specific parameters";
                      }
            container ipv6 {
             leaf address-allocation-type {
              type string;
              description
               "Defines how addresses are allocated.
                Need to be detailled further.";
              }
             leaf subnet-prefix {
              type inet:ipv6-prefix;
              description
              "Interco subnet.";
                      }
              description
                      "IPv6 specific parameters";
              }
            container oam {
                     container 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 holdtime
                  expressed
```

```
                in msec.";
              }
             }
             description
             "Choice for holdtime flavor.";}
             description
              "Container for BFD.";}
               description
                           "Define the OAM used on the connection.";}
                list routing-protocols {
                 key type;
                  leaf type {
                   type identityref {
                   base l3vpn-svc:routing-protocol-type;
                   }
                                  description
                                  "Type of routing protocol.";
                  }

                  container ospf {
                   when "type = 'ospf'" {
                    description
                     "Only applies
                     when protocol is OSPF.";
                     }
                    leaf-list address-family {
                     type identityref {
                                    base rt:address-family;
                     }
                     description
                      "Address family to be activated.";
                     }
                        leaf area-address {
                            type yang:dotted-quad;
                            description
                             "Area address.";
                        }
                        leaf metric {
                            type uint16;
                            description
                             "Metric of PE-CE link.";
                        }
                        list sham-link {
                            key target-site;

                            leaf target-site {
                                type leafref {
                                 path "../../../../../../"
```

```
                        +"../sites/site-id";
                       }
                       description
                        "Target site for the sham link
                         connection.";
                   }
                   leaf metric {
                       type uint16;
                       description
                        "Metric of the sham link.";
                   }
                   description
                    "Creates a shamlink with another
                    site";
               }
               description
                "OSPF specific configuration.";
           }

           container bgp {
               when "type = 'bgp'" {
                   description
                    "Only applies when
                    protocol is BGP.";
               }
               leaf-list address-family {
                   type identityref {
                       base rt:address-family;
                   }
                   description
                    "Address family to be activated.";
               }
               description
                "BGP specific configuration.";
           }
           container static {
               when "type = 'static'" {
                   description
                    "Only applies when protocol
                    is static.";
               }
               leaf-list address-family {
                   type identityref {
                       base rt:address-family;
                   }
                   description
                    "Address family to be activated.";
               }
```

```
                    description
                     "Static routing
                     specific configuration.";
                }
                container rip {
                    when "type = 'rip'" {
                        description
                         "Only applies when
                         protocol is RIP.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Address family to be
                         activated.";
                    }

                    description
                     "RIP routing specific
                     configuration.";
                }


                container vrrp {
                    when "type = 'vrrp'" {
                        description
                         "Only applies when
                         protocol is VRRP.";
                    }
                    leaf-list address-family {
                        type identityref {
                            base rt:address-family;
                        }
                        description
                         "Address family to be activated.";
                    }
                    description
                     "VRRP routing specific configuration.";
                }
                description
                 "List of routing protocols used
                 on the site.
                 Need to be augmented.";
            }
            description
             "Defines connection parameters.";
```

```
              }
             }
            }

            container service{
             description
             "Service parameters on the attachement.";
             uses l3vpn-svc:site-service-basic;
             container qos{
              description
              "TBD.";
              container qos-classification-policy {
               description
                   "QoS configuration";
           list rules {
                    key id;
                    description
                    "list of qos rules.";
                    leaf id {
             type uint16;
             description
             "ID of the rule.";
            }
                    container match-flow{
                    description
                    "container of match flow.";
                    leaf dest-mac-address{
                     type yang:mac-address;
                     description
                     "destination mac address.";
                    }

                    leaf src-mac-address{
                     type yang:mac-address;
                     description
                     "source mac address.";
                    }

                    leaf local-label{
                     type string;
                     description
                     "local label.";
                    }

                    leaf remote-label{
                     type string;
                     description
                     "remote label.";
```

```
                }

               leaf dot1q-vlan-bitmap {
                   type string;
            mandatory true;
            description "Dot1Q Vlan Bitmap." ;
        }

        leaf qinq-svlan-bitmap {
            type string;
                        mandatory true;
            description "QinQ svlan Bitmap." ;
        }

        leaf qinq-cvlan-bitmap {
            type string;
            mandatory true;
            description "QinQ cvlan Bitmap." ;
        }
        leaf target-class-id {
          type string;
          description
           "Identification of the
            class of service.
            This identifier is internal to
             the administration."; }
                }
              leaf std-qos-profile {
                  type string;
                  description
                   "QoS profile to be used";
              }
              container custom-qos-profile {
                  list class {
                      key class-id;

                      leaf class-id {
                          type string;
                          description
                           "Identification of the
                            class of service.
                            This identifier is internal to
                            the administration.";
                      }
                      leaf rate-limit {
                          type uint8;
                          units percent;
                          description
```

```
                            "To be used if class must
                            be rate
                            limited. Expressed as
                            percentage of the svc-bw.";
                        }
                        leaf priority-level {
                            type uint8;
                            description
                            "Defines the level of the
                            class in
                            term of priority queueing.
                             The higher the level is the
                             higher
                             is the priority.";
                        }
                        leaf guaranteed-bw-percent {
                            type uint8;
                            units percent;
                            description
                            "To be used to define the
                            guaranteed
                            BW in percent of the svc-bw
                            available at the priority-level.";
                        }
                        description
                         "List of class of services.";
                    }
                    description
                     "Custom qos profile.";
                }
            }
               }
              }
              uses l3vpn-svc:site-service-protection;
              uses l3vpn-svc:site-service-mpls;
              uses l3vpn-svc:site-service-multicast;
             }
            }

      }
<CODE ENDS>
```

6. Security Considerations

TBC.

7. IANA Considerations

TBC.

8. Conclusion

This document intends to trigger a discussion at IETF 94 meeting in Yokohama on other VPN service modeling. It uses L2VPN service model as an example to explore how L3VPN service model can be used as basis to define other type of VPN service models such as Cloud VPN service model, OTT VPN service model, Hybrid VPN service model. Right now L3VPN service model defined in L3SM WG follows modularity approach and has been defined in more extensible way, therefore other VPN service model can reuse building blocks defined in L3SM service model without need of reinventing a new wheel. However L3VPN service model can not be directly extended to other VPN service model since it include L3VPN specific aspect, e.g., IP connection, QoS filter and Routing filter that are applied to IP network. Therefore we can not use L3VPN service model structure as a common structure for other VPN service models.

9. Acknowledgements

The authors would like to thank Zitao Wang for the very fruitful discussions and useful suggestions in the initial version.

10. Normative References

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

[RFC3809] Nagarajan, A., Ed., "Generic Requirements for Provider Provisioned Virtual Private Networks (PPVPN)", RFC 3809, DOI 10.17487/RFC3809, June 2004, <http://www.rfc-editor.org/info/rfc3809>.

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

Authors' Addresses

Chongfeng Xie
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
China

Email: xiechf@ctbri.com.cn

Aijun Wang
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
China

Email: wangaj@ctbri.com.cn

Roland Schott
Deutsche Telekom
Deutsche-Telekom-Allee 7
Darmstadt 64295
Germany

Email: Roland.Schott@telekom.de

Dacheng Zhang

Email: dacheng.zhang@gmail.com