opsawg
Internet-Draft
Intended status: Standards Track
Expires: March 6, 2021

S. Barguil
O. Gonzalez de Dios, Ed.
Telefonica
M. Boucadair, Ed.
Orange
Q. Wu
Huawei

September 2, 2020

A Layer 2/3 VPN Common YANG Model
draft-ietf-opsawg-vpn-common-00

Abstract

This document defines a common YANG module that is meant to be reused by various VPN-related modules such as Layer 3 VPN Service Model, Layer 2 VPN Service Model, Layer 3 VPN Network Model, and Layer 2 VPN Network Model.

Editorial Note (To be removed by RFC Editor)

Please update these statements within the document with the RFC number to be assigned to this document:

- "This version of this YANG module is part of RFC XXXX;"
- "RFC XXXX: A Layer 2/3 VPN Common YANG Model";
- reference: RFC XXXX

Also, please update the "revision" date of the YANG module.

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 https://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 March 6, 2021.

Table of Contents

1. Introduction

Various VPN-related YANG data modules were specified by the IETF (e.g., Layer 3 VPN Service Model (L3SM) [RFC8299] or Layer 2 VPN Service Model (L2SM) [RFC8466]). Others are also being specified (e.g., Layer 3 VPN Network Model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm] or Layer 2 VPN Network Model (L2NM) [I-D.ietf-opsawg-l2nm]). These modules have data nodes and structures that are present in almost all these models or a subset of them. An example of such data nodes is depicted in Figure 1.

```
module: ietf-l2vpn-ntw
   +--rw vpn-services
       +--rw vpn-service* [vpn-id]
             +--rw vpn-id                      svc-id
             +--rw vpn-svc-type?               identityref
             +--rw customer-name?              string
             +--rw svc-topo?                   identityref
             +-rw service-status
             |  +-rw admin
             |  |  +-rw status?      operational-type
             |  |  +-rw timestamp?   yang:date-and-time
             |  +-ro ops
             |     +-ro status?      operational-type
             |     +-ro timestamp?   yang:date-and-time
             |  ...

module: ietf-l3vpn-ntw
   +--rw vpn-services
       +--rw vpn-service* [vpn-id]
             +-rw service-status
             |  +-rw admin
             |  |  +-rw status?      operational-type
             |  |  +-rw timestamp?   yang:date-and-time
             |  +-ro ops
             |     +-ro status?      operational-type
             |     +-ro timestamp?   yang:date-and-time
             +--rw vpn-id                   l3vpn-svc:svc-id
             +--rw l3sm-vpn-id?             l3vpn-svc:svc-id
             +--rw customer-name?           string
             +--rw vpn-service-topology?    identityref
             +--rw description?             string
             |  ...
```

Figure 1: Example of Common Data Nodes in Both L2NM/L3NM

In order to avoid data nodes duplication and to ease passing data among layers (service layer to network layer and vice versa), early versions of the L3NM reused many of the data nodes that are defined in the L3SM [RFC8299]. Nevertheless, that approach was abandoned because that design was interpreted as if the deployment of L3NM depends on L3SM, while this is not required. For example, a Service Provider may decide to use the L3NM to build its L3VPN services without exposing the L3SM.

Likewise, early versions of the L2NM reused many of the data nodes that are defined in both L2SM and L3NM. An example of L3NM groupings reused in L3NM is shown in Figure 2. This data nodes reuse was

interpreted as if the deployment of L2NM requires both L3NM; which is not required.

```
module ietf-l2vpn-ntw {
 ...
  import ietf-l3vpn-ntw {
    prefix l3vpn-ntw;
    reference
      "RFC NNNN: A Layer 3 VPN Network YANG Model";
  }
  ...
  container l2vpn-ntw {
    ...
    container vpn-services {
      list vpn-service {
        ...
        uses l3vpn-ntw:service-status;
        uses l3vpn-ntw:svc-transport-encapsulation;
        ...
      }
    }
    ...
  }
}
```

Figure 2: Excerpt from the L2NM YANG Module

To avoid the issues discussed above, this document defines a common YANG module that is meant to be reused by various VPN-related modules such as Layer 3 VPN Service Model (L3SM) [RFC8299], Layer 2 VPN Service Model (L2SM) [RFC8466], Layer 3 VPN Network Model (L3NM) [I-D.ietf-opsawg-l3sm-l3nm], and Layer 2 VPN Network Model (L2NM) [I-D.ietf-opsawg-l2nm]: "ietf-vpn-common" (Section 4).

The "ietf-vpn-common" module includes a set of identities, types, and groupings that are meant to be reused by other VPN-related YANG modules independently of their layer (e.g., Layer 2, Layer 3) and the type of the module (e.g., network model, service model) including future revisions of existing models (e.g., L3SM [RFC8299] or L3SM [RFC8466]).

The approach that is followed for building the common module (Section 4) is to first extract data nodes that are common for both L3NM and L3SM; these data nodes are then filtered out against Layer 2 modules. All the common groupings are called, for example, in the L3NM module defined in [I-D.ietf-opsawg-l3sm-l3nm].

2. Terminology

The terminology for describing YANG modules is defined in [RFC7950].

The meaning of the symbols in tree diagrams is defined in [RFC8340].

3. Description of the VPN Common YANG Module

The "ietf-vpn-common" contains the following reusable groupings and identities:

Groupings:

- vpn-description:

  A YANG grouping that provides common administrative VPN information such as a name, a textual description, and a customer name.

- vpn-profile-cfg:

  A YANG grouping that defines a set of profiles (encryption, routing, forwarding) valid for any L2/L3 VPN.

- status-timestamp:

  A YANG grouping that defines operational and administrative updates of a component.

- service-status:

  A YANG grouping that defines the administrative and operational status of a component. The grouping can be applied to the whole service or an endpoint.

- svc-transport-encapsulation:

  A YANG grouping that defines the type of underlay transport for a VPN service.

- rt-rd:

  A YANG grouping that defines the set of route targets to match for import and export routes to/from VRF.

- vpn-node-group:

  A YANG grouping that is used to group vpn-network-access.

Identities:

- bw-direction: Identity for the bandwidth direction.
- qos-profile-direction: Base identity for QoS profile direction.
- customer-application: Base identity for customer application.
- ie-type: Identity for Import-Export routing profiles.
- site-network-access-type: Base identity for site-network-access type.
- operational-status: Base identity for the operational status.
- administrative-status: Base identity for administrative status.
- encapsulation-type: Base identity for encapsulation type.
- tag-type: Base identity from which all tag types are derived.
- protocol-type: Base identity for Protocol Type.
- vpn-topology: Base identity for VPN topology.
- role: Base identity for site or node type.
- vpn-signaling-type: Identity of VPN signaling types
- service-type: Identity of service type.
- vxlan-peer-mode: Base identity for the VXLAN peer mode.
- multicast-gp-address-mapping: Identity for multicast group mapping type.
- multicast-tree-type: Base identity for multicast tree type.
- multicast-rp-discovery-type: Base identity for RP discovery type.

The tree diagram of the "ietf-vpn-common" module that depicts the common groupings is provided in Figure 3. More descriptions of these groupings are provided in the description statements in Section 4.

```
module: ietf-vpn-common

  grouping vpn-description
    +-- vpn-id?            vpn-common:vpn-id
```

```
  +-- vpn-name?          string
  +-- vpn-description?   string
  +-- customer-name?     string
grouping vpn-profile-cfg
  +-- valid-provider-identifiers
     +-- cloud-identifier* [id] {cloud-access}?
     |  +-- id?   string
     +-- encryption-profile-identifier* [id]
     |  +-- id?   string
     +-- qos-profile-identifier* [id]
     |  +-- id?   string
     +-- bfd-profile-identifier* [id]
     |  +-- id?   string
     +-- forwarding-profile-identifier* [id]
     |  +-- id?   string
     +-- routing-profile-identifier* [id]
        +-- id?   string
grouping status-timestamp
  +-- status?         identityref
  +-- last-updated?   yang:date-and-time
grouping service-status
  +-- status
     +-- admin-status
     |  +-- status?         identityref
     |  +-- last-updated?   yang:date-and-time
     +--ro oper-status
        +--ro status?         identityref
        +--ro last-updated?   yang:date-and-time
grouping svc-transport-encapsulation
  +-- underlay-transport
     +-- type*   identityref
grouping rt-rd
  +-- rd?            union
  +-- vpn-targets
     +-- vpn-target* [id]
     |  +-- id?                  int8
     |  +-- route-targets* [route-target]
     |  |  +-- route-target?   rt-types:route-target
     |  +-- route-target-type    rt-types:route-target-type
     +-- vpn-policies
        +-- import-policy?   string
        +-- export-policy?   string
grouping vpn-route-targets
  +-- vpn-target* [id]
  |  +-- id?                  int8
  |  +-- route-targets* [route-target]
  |  |  +-- route-target?   rt-types:route-target
  |  +-- route-target-type    rt-types:route-target-type
```

```
          +-- vpn-policies
             +-- import-policy?   string
             +-- export-policy?   string
        grouping vpn-node-group
          +-- groups
             +-- group* [group-id]
                +-- group-id?   string
```

Figure 3: VPN Common Tree

4. Layer 2/3 VPN Common Module

This module uses types defined in [RFC6991] and [RFC8294].

Editor's Note: RFCs cited in the reference statements will be added to the References Section in future versions.

```
<CODE BEGINS>  file "ietf-vpn-common@2020-07-13.yang"
module ietf-vpn-common {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-vpn-common";
  prefix vpn-common;

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }
  import ietf-routing-types {
    prefix rt-types;
    reference
      "RFC 8294: Common YANG Data Types for the Routing Area";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "Section 3 of RFC 6991";
  }

  organization
    "IETF OPSA (Operations and Management Area) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Editor:  Samier Barguil
              <mailto:samier.barguilgiraldo.ext@telefonica.com>
```

```
      Editor:  Oscar Gonzalez de Dios
               <mailto:oscar.gonzalezdedios@telefonica.com>

      Editor:  Mohamed Boucadair
               <mailto:mohamed.boucadair@orange.com>

      Author:  Qin Wu
               <mailto:bill.wu@huawei.com>";
   description
     "This YANG module defines a common module that is meant
      to be reused by various VPN-related modules (e.g.,
      Layer 3 VPN Service Model (L3SM), Layer 2 VPN Service
      Model (L2SM), Layer 3 VPN Network Model (L3NM), Layer 2
      VPN Network Model (L2NM)).

      Copyright (c) 2020 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject to
      the license terms contained in, the Simplified BSD License set
      forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

      This version of this YANG module is part of RFC XXXX
      (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
      for full legal notices.";

   revision 2020-07-13 {
     description
       "Initial revision.";
     reference
       "RFC XXXX: A Layer 2/3 VPN Common YANG Model";
   }

   /* Features */

   feature cloud-access {
     description
       "Indicates support of the VPN to connect to a Cloud
        Service Provider (CSP).";
   }

   feature lag-interface {
     description
       "Indicates the support of Link aggregation between
        Site Network Accesses. ";
```

```
  }

  feature site-diversity {
    description
      "Indicates the site diversity in the customer premises.";
  }

  feature dot1q {
    description
      "This feature indicates the support of
       the 'dot1q' encapsulation.";
  }

  feature qinq {
    description
      "This feature indicates the support of
       the 'qinq' encapsulation.";
  }

  feature vxlan {
    description
      "This feature indicates the support of
       the 'vxlan' encapsulation.";
  }

  feature qinany {
    description
      "This feature indicates the support of
       the 'qinany' encapsulation.";
  }

  feature multicast {
    description
      "Indicates multicast capabilities support in a VPN.";
  }

  feature ipv4 {
    description
      "Indicates IPv4 support in a VPN.";
  }

  feature ipv6 {
    description
      "Indicates IPv6 support in a VPN.";
  }

  feature carrierscarrier {
    description
```

```
      "Indicates support of Carrier-of-Carrier VPNs.";
  }

  feature extranet-vpn {
    description
      "Indicates support of extranet VPNs.";
  }

  feature fast-reroute {
    description
      "Indicates support of Fast Reroute (FRR).";
  }

  feature qos {
    description
      "Indicates support of classes of services (CoSes).";
  }

  feature encryption {
    description
      "Indicates support of encryption.";
  }

  feature bfd {
    description
      "Indicates support of BFD.";
  }

  feature bearer-reference {
    description
      "Indicates support of the 'bearer-reference' access
       constraint.";
  }

  feature input-bw {
    description
      "This feature indicates the support of
       the 'input-bw' limit.";
  }

  /* Typedef */

  typedef vpn-id {
    type string;
    description
      "Defines an identifier that is used as
       a service identifier, for example.";
  }
```

```
   typedef address-family {
     type enumeration {
       enum ipv4 {
         description
           "IPv4 address family.";
       }
       enum ipv6 {
         description
           "IPv6 address family.";
       }
     }
     description
       "Defines a type for the address family.";
   }

   /* Identities */

   identity bw-direction {
     description
       "Identity for the bandwidth direction.";
   }

   identity input-bw {
     base bw-direction;
     description
       "Identity for the input bandwidth.";
   }

   identity output-bw {
     base bw-direction;
     description
       "Identity for the output bandwidth.";
   }

   identity qos-profile-direction {
     description
       "Base identity for QoS profile direction.";
   }

   identity site-to-wan {
     base qos-profile-direction;
     description
       "Identity for Site-to-WAN direction.";
   }

   identity wan-to-site {
     base qos-profile-direction;
     description
```

```
      "Identity for WAN-to-Site direction.";
  }

  identity both {
    base qos-profile-direction;
    description
      "Identity for both WAN-to-Site direction
       and Site-to-WAN direction.";
  }

  identity customer-application {
    description
      "Base identity for customer application.";
  }

  identity web {
    base customer-application;
    description
      "Identity for Web application (e.g., HTTP, HTTPS).";
  }

  identity mail {
    base customer-application;
    description
      "Identity for mail application.";
  }

  identity file-transfer {
    base customer-application;
    description
      "Identity for file transfer application (e.g., FTP, SFTP).";
  }

  identity database {
    base customer-application;
    description
      "Identity for database application.";
  }

  identity social {
    base customer-application;
    description
      "Identity for social-network application.";
  }

  identity games {
    base customer-application;
    description
```

```
      "Identity for gaming application.";
  }

  identity p2p {
    base customer-application;
    description
      "Identity for peer-to-peer application.";
  }

  identity network-management {
    base customer-application;
    description
      "Identity for management application
       (e.g., Telnet, syslog, SNMP).";
  }

  identity voice {
    base customer-application;
    description
      "Identity for voice application.";
  }

  identity video {
    base customer-application;
    description
      "Identity for video conference application.";
  }

  identity embb {
    base customer-application;
    description
      "Identity for an enhanced Mobile Broadband (eMBB)
       application.  Note that an eMBB application demands
       network performance with a wide variety of
       characteristics, such as data rate, latency,
       loss rate, reliability, and many other parameters.";
  }

  identity urllc {
    base customer-application;
    description
      "Identity for an Ultra-Reliable and Low Latency
       Communications (URLLC) application.  Note that a
       URLLC application demands network performance
       with a wide variety of characteristics, such as latency,
       reliability, and many other parameters.";
  }
```

```
   identity mmtc {
     base customer-application;
     description
       "Identity for a massive Machine Type
        Communications (mMTC) application.  Note that an
        mMTC application demands network performance
        with a wide variety of characteristics, such as data
        rate, latency, loss rate, reliability, and many
        other parameters.";
   }

   identity ie-type {
     description
       "Defines Import-Export routing profiles.
        Those profiles can be reused between VPN nodes.";
   }

   identity import {
     base ie-type;
     description
       "Import a routing profile.";
   }

   identity export {
     base ie-type;
     description
       "Export a routing profile.";
   }

   identity import-export {
     base ie-type;
     description
       "Import/Export a routing profile.";
   }

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

   identity point-to-point {
     base site-network-access-type;
     description
       "Identity for point-to-point connection.";
   }

   identity multipoint {
     base site-network-access-type;
```

```
    description
      "Identity for multipoint connection.
       Example: Ethernet broadcast segment.";
  }

  identity pseudowire {
    base site-network-access-type;
    description
      "Identity for pseudowire connections.";
  }

  identity loopback {
    base site-network-access-type;
    description
      "Identity for loopback connections.";
  }

  identity operational-status {
    description
      "Base identity for the operational status.";
  }

  identity operational-state-up {
    base operational-status;
    description
      "Operational status is UP/Enabled.";
  }

  identity operational-state-down {
    base operational-status;
    description
      "Operational status is DOWN/Disabled.";
  }

  identity operational-state-unknown {
    base operational-status;
    description
      "Operational status is UNKNOWN.";
  }

  identity administrative-status {
    description
      "Base identity for administrative status.";
  }

  identity administrative-state-up {
    base administrative-status;
    description
```

```
      "Administrative status is UP/Enabled.";
  }

  identity administrative-state-down {
    base administrative-status;
    description
      "Administrative status is DOWN/Disabled.";
  }

  identity administrative-state-testing {
    base administrative-status;
    description
      "Administrative status is up for testing purposes.";
  }

  identity administrative-state-pre-deployment {
    base administrative-status;
    description
      "Administrative status is pre-deployment phase.";
  }

  identity encapsulation-type {
    description
      "Base identity for encapsulation type.";
  }

  identity priority-tagged {
    base encapsulation-type;
    description
      "Identity for the priority-tagged interface.";
  }

  identity dot1q {
    base encapsulation-type;
    description
      "This identity indicates the support of
       the 'dot1q' encapsulation.";
  }

  identity qinq {
    base encapsulation-type;
    description
      "This identity indicates the support of
       the 'qinq' encapsulation.";
  }

  identity qinany {
    base encapsulation-type;
```

```
    description
      "This identity indicates the support of
       the 'qinany' encapsulation.";
  }

  identity vxlan {
    base encapsulation-type;
    description
      "This identity indicates the support of
       the 'vxlan' encapsulation.";
  }

  identity ethernet-type {
    base encapsulation-type;
    description
      "Identity for encapsulation type.";
  }

  identity vlan-type {
    base encapsulation-type;
    description
      "Identity for VLAN encapsulation.";
  }

  identity untagged-int {
    base encapsulation-type;
    description
      "Identity for Ethernet type.";
  }

  identity tagged-int {
    base encapsulation-type;
    description
      "Identity for the VLAN type.";
  }

  identity lag-int {
    base encapsulation-type;
    description
      "Identity for the VLAN type.";
  }

  identity tag-type {
    description
      "Base identity from which all tag types are derived.";
  }

  identity c-vlan {
```

```
    base tag-type;
    description
      "A CVLAN tag, normally using the 0x8100 Ethertype.";
  }

  identity s-vlan {
    base tag-type;
    description
      "An SVLAN tag.";
  }

  identity c-s-vlan {
    base tag-type;
    description
      "Using both a CVLAN tag and an SVLAN tag.";
  }

  identity protocol-type {
    description
      "Base identity for Protocol Type.";
  }

  identity gre {
    base protocol-type;
    description
      "GRE encapsulation.";
    reference
      "RFC 1701: Generic Routing Encapsulation (GRE)
       RFC 1702: Generic Routing Encapsulation over IPv4 networks
       RFC 7676: IPv6 Support for Generic Routing Encapsulation
                 (GRE)";
  }

  identity ldp {
    base protocol-type;
    description
      "Transport based on LDP.";
    reference
      "RFC 3086: LDP Specification";
  }

  identity sr {
    base protocol-type;
    description
      "Transport based on SR.";
    reference
      "RFC 8660: Segment Routing with the MPLS Data Plane
       RFC 8663: MPLS Segment Routing over IP
```

```
        RFC 8754: IPv6 Segment Routing Header (SRH)";
    }

    identity sr-te {
      base protocol-type;
      description
        "Transport based on SR-TE.";
      reference
        "RFC 8426: Recommendations for RSVP-TE and Segment Routing (SR)
         Label Switched Path (LSP) Coexistence";
    }

    identity rsvp-te {
      base protocol-type;
      description
        "Transport based on RSVP-TE.";
      reference
        "RFC 2205: Resource ReSerVation Protocol (RSVP) --
         Version 1 Functional Specification";
    }

    identity bgp-lu {
      base protocol-type;
      description
        "Transport based on BGP-LU.";
    }

    identity unknown {
      base protocol-type;
      description
        "Not known at this stage.";
    }

    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-and-Spoke VPN topology.";
```

```
   }

   identity hub-spoke-disjoint {
     base vpn-topology;
     description
       "Identity for Hub-and-Spoke VPN topology
        where Hubs cannot communicate with each other.";
   }

   identity custom {
     base vpn-topology;
     description
       "Identity for CUSTOM VPN topology
        where Hubs can act as Spoke for certain part of
        the network or Spokes as Hubs.";
   }

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

   identity any-to-any-role {
     base role;
     description
       "VPN-Node in an any-to-any IP VPN.";
   }

   identity spoke-role {
     base role;
     description
       "VPN-Node acting as a Spoke IP VPN.";
   }

   identity hub-role {
     base role;
     description
       "VPN-Node acting as a Hub IP VPN.";
   }

   identity custom-role {
     base role;
     description
       "VPN-Node with custom or complex role in the VPN.";
   }

   identity vpn-signaling-type {
     description
```

```
      "Identity of VPN signaling types";
  }

  identity l2vpn-bgp {
    base vpn-signaling-type;
    description
      "Identity of l2vpn-bgp";
  }

  identity evpn-bgp {
    base vpn-signaling-type;
    description
      "Identity of evpn-bgp";
  }

  identity t-ldp {
    base vpn-signaling-type;
    description
      "Identity of t-ldp.";
  }

  identity h-vpls {
    base vpn-signaling-type;
    description
      "Identity for h-vpls";
  }

  identity l2tp {
    base vpn-signaling-type;
    description
      "Identity of l2tp.";
  }

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

  identity l3vpn {
    base service-type;
    description
      "Identity of L3VPN service.";
  }

  identity vpws {
    base service-type;
    description
      "Point-to-point Virtual Private Wire Service (VPWS)
```

```
       service type.";
  }

  identity pwe3 {
    base service-type;
    description
      "Pseudowire Emulation Edge to Edge (PWE3) service type.";
  }

  identity ldp-l2tp-vpls {
    base service-type;
    description
      "LDP-based or L2TP-based multipoint Virtual Private LAN
       Service (VPLS) service type.  This VPLS uses LDP-signaled
       Pseudowires or L2TP-signaled Pseudowires.";
  }

  identity bgp-vpls {
    base service-type;
    description
      "BGP-based multipoint VPLS service type.  This VPLS uses a
       BGP control plane.";
    reference
      "RFC4761:  Virtual Private LAN Service (VPLS) Using
                 BGP for Auto-Discovery and Signaling
       RFC 6624: Layer 2 Virtual Private Networks Using BGP for
                 Auto-Discovery and Signaling";
  }

  identity vpws-evpn {
    base service-type;
    description
      "VPWS service type using Ethernet VPNs (EVPNs).";
    reference
      "RFC 7432: BGP MPLS-Based Ethernet VPN";
  }

  identity pbb-evpn {
    base service-type;
    description
      "PBB EVPN.";
  }

  identity vxlan-peer-mode {
    description
      "Base identity for the VXLAN peer mode.";
  }
```

```
   identity static-mode {
     base vxlan-peer-mode;
     description
       "Identity for VXLAN access in the static mode.";
   }

   identity bgp-mode {
     base vxlan-peer-mode;
     description
       "Identity for VXLAN access by BGP EVPN learning.";
   }

   identity multicast-gp-address-mapping {
     description
       "Identity for multicast group mapping type.";
   }

   identity static-mapping {
     base multicast-gp-address-mapping;
     description
       "Identity for static mapping, i.e., attach the interface
        to the multicast group as a static member.";
   }

   identity dynamic-mapping {
     base multicast-gp-address-mapping;
     description
       "Identity for dynamic mapping, i.e., an interface was added
        to the multicast group as a result of snooping.";
   }

   identity multicast-tree-type {
     description
       "Base identity for multicast tree type.";
   }

   identity ssm-tree-type {
     base multicast-tree-type;
     description
       "Identity for SSM tree type.";
   }

   identity asm-tree-type {
     base multicast-tree-type;
     description
       "Identity for ASM tree type.";
   }
```

```
   identity bidir-tree-type {
     base multicast-tree-type;
     description
       "Identity for bidirectional tree type.";
   }

   identity multicast-rp-discovery-type {
     description
       "Base identity for RP discovery type.";
   }

   identity auto-rp {
     base multicast-rp-discovery-type;
     description
       "Base identity for Auto-RP discovery type.";
   }

   identity static-rp {
     base multicast-rp-discovery-type;
     description
       "Base identity for static type.";
   }

   identity bsr-rp {
     base multicast-rp-discovery-type;
     description
       "Base identity for BSR discovery type.";
   }

   /* Grouping */

   grouping vpn-description {
     leaf vpn-id {
       type vpn-common:vpn-id;
       description
         "VPN identifier.
          This identifier has a local meaning.";
     }
     leaf vpn-name {
       type string;
       description
         "A name used to refer to the VPN.";
     }
     leaf vpn-description {
       type string;
       description
         "Textual description of a VPN service.";
     }
```

```
    leaf customer-name {
      type string;
      description
        "Name of the customer that actually uses the VPN service.";
    }
    description
      "Provides common VPN information.";
  }

  grouping vpn-profile-cfg {
    container valid-provider-identifiers {
      list cloud-identifier {
        if-feature "cloud-access";
        key "id";
        leaf id {
          type string;
          description
            "Identification of cloud service.
             Local administration meaning.";
        }
        description
          "List for Cloud Identifiers.";
      }
      list encryption-profile-identifier {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the SP encryption profile
             to be used.  Local administration meaning.";
        }
        description
          "List for encryption profile identifiers.";
      }
      list qos-profile-identifier {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the QoS Profile to be used.
             Local administration meaning.";
        }
        description
          "List for QoS Profile Identifiers.";
      }
      list bfd-profile-identifier {
        key "id";
        leaf id {
```

```
          type string;
          description
            "Identification of the SP BFD Profile to be used.
             Local administration meaning.";
        }
        description
          "List for BFD Profile identifiers.";
      }
      list forwarding-profile-identifier {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the Forwrding Profile Filter to be used.
             Local administration meaning.";
        }
        description
          "List for Forwrding Profile identifiers.";
      }
      list routing-profile-identifier {
        key "id";
        leaf id {
          type string;
          description
            "Identification of the routing Profile to be used
             by the routing-protocols within sites, vpn-
             network-accesses or vpn-nodes for refering
             vrf-import/export policies.
             This identifier has a local meaning.";
        }
        description
          "List for Routing Profile Identifiers.";
      }
      nacm:default-deny-write;
      description
        "Container for Valid Provider Identifies.";
    }
    description
      "Grouping for VPN Profile configuration.";
  }

  grouping status-timestamp {
    leaf status {
      type identityref {
        base operational-status;
      }
      description
        "Operations status";
```

```
    }
    leaf last-updated {
      type yang:date-and-time;
      description
        "Indicates the actual date and time of the service
         status change.";
    }
    description
      "This grouping defines some operational
       parameters for the service.";
  }

  grouping service-status {
    container status {
      container admin-status {
        leaf status {
          type identityref {
            base administrative-status;
          }
          description
            "Administrative service status.";
        }
        leaf last-updated {
          type yang:date-and-time;
          description
            "Indicates the actual date and time of the service
             status change.";
        }
        description
          "Administrative service status.";
      }
      container oper-status {
        config false;
        uses status-timestamp;
        description
          "Operational service status.";
      }
      description
        "Service status.";
    }
    description
      "Service status grouping.";
  }

  grouping svc-transport-encapsulation {
    container underlay-transport {
      leaf-list type {
        type identityref {
```

```
          base protocol-type;
        }
        ordered-by user;
        description
          "Protocols used to deliver a VPN service.";
      }
      description
        "Container for the Transport underlay.";
    }
    description
      "This grouping defines the type of underlay transport
       for VPN service.";
  }

  grouping rt-rd {
    leaf rd {
      type union {
        type rt-types:route-distinguisher;
        type empty;
      }
      description
        "Route distinguisher value. If this leaf has not been
         configured, the server will auto-assign a route
         distinguisher value and use that value operationally.
         This calculated value is available in the operational
         state.

         Use the empty type to indicate RD has no value and
         is not to be aouto-assigned.";
    }
    container vpn-targets {
      description
        "Set of route-targets to match for import and export routes
         to/from VRF";
      uses vpn-route-targets;
    }
    description
      "Grouping for RT and RD.";
  }

  grouping vpn-route-targets {
    description
      "A grouping that specifies Route Target import-export rules
       used in a BGP-enabled VPN.";
    list vpn-target {
      key "id";
      leaf id {
        type int8;
```

```
        description
          "Identifies each VPN Target";
      }
      list route-targets {
        key "route-target";
        leaf route-target {
          type rt-types:route-target;
          description
            "Route Target value";
        }
        description
          "List of Route Targets.";
      }
      leaf route-target-type {
        type rt-types:route-target-type;
        mandatory true;
        description
          "Import/export type of the Route Target.";
      }
      description
        "L3VPN route targets. AND/OR Operations are available
         based on the RTs assigment.";
    }
    reference
      "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs)
       RFC4664: Framework for Layer 2 Virtual Private Networks
       (L2VPNs)";
    container vpn-policies {
      description
        "VPN policies";
      leaf import-policy {
        type string;
        description
          "Defines the import policy.";
      }
      leaf export-policy {
        type string;
        description
          "Defines the export policy.";
      }
    }
  }

  grouping vpn-node-group {
    container groups {
      list group {
        key "group-id";
        leaf group-id {
```

```
          type string;
          description
            "Group-id the vpn-node belongs to.";
        }
        description
          "List of group-ids.";
      }
      description
        "Groups the vpn node and network access belongs to.";
    }
    description
      "Grouping definition to assign
       group-ids to group or network access.";
  }
}
<CODE ENDS>
```

5. Security Considerations

The YANG modules specified in this document define schemas for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

The "ietf-vpn-common" module defines a set of identities, types, and groupings. These nodes are intended to be reused by other YANG modules. As such, the module does not expose by itself any data nodes which are writable, contain read-only state, or RPCs. As such, there are no additional security issues to be considered relating to the "ietf-vpn-common" module.

6. IANA Considerations

This document requests IANA to register the following URI in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-vpn-common
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG module in the "YANG Module Names" subregistry [RFC6020] within the "YANG Parameters" registry.

```
name: ietf-vpn-common
namespace: urn:ietf:params:xml:ns:yang:ietf-vpn-common
maintained by IANA: N
prefix: vpn-common
reference: RFC XXXX
```

7. Contributors

Italo Busi
Huawei Technologies
Email: Italo.Busi@huawei.com

Luis Angel Munoz
Vodafone
Email: luis-angel.munoz@vodafone.com

Victor Lopez Alvarez
Telefonica
Email: victor.lopezalvarez@telefonica.com

8. References

8.1. Normative References

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

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://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, <https://www.rfc-editor.org/info/rfc6241>.

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

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, DOI 10.17487/RFC8294, December 2017, <https://www.rfc-editor.org/info/rfc8294>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Informative References

[I-D.ietf-opsawg-l2nm] barguil, s., Dios, O., Boucadair, M., Munoz, L., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", draft-ietf-opsawg-l2nm-00 (work in progress), July 2020.

[I-D.ietf-opsawg-l3sm-l3nm] barguil, s., Dios, O., Boucadair, M., Munoz, L., and A. Aguado, "A Layer 3 VPN Network YANG Model", draft-ietf-opsawg-l3sm-l3nm-03 (work in progress), April 2020.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/info/rfc8299>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <https://www.rfc-editor.org/info/rfc8466>.

Authors' Addresses

Samier Barguil
Telefonica
Madrid
ES

Email: samier.barguilgiraldo.ext@telefonica.com

Oscar Gonzalez de Dios (editor)
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Mohamed Boucadair (editor)
Orange
France

Email: mohamed.boucadair@orange.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com