Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2021

J. Dong
X. Wei
Q. Wu
Huawei
M. Boucadair
Orange
A. Liu
Tecent

July 12, 2020

A YANG Data Model for Layer 2 Network Topologies
draft-ietf-i2rs-yang-l2-network-topology-15

Abstract

This document defines a YANG data model for Layer 2 network topologies. In particular, this data model augments the generic network and network topology data models with Layer 2 specific topology attributes.

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 YANG Data Model for Layer 2 Network Topologies";
- reference: RFC XXXX

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 January 13, 2021.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

[RFC8345] defines the YANG [RFC6020] [RFC7950] data models of the abstract (generic) network and network topology. Such models can be augmented with technology-specific details to build more specific topology models.

This document defines the YANG data model for Layer 2 (L2) network topologies by augmenting the generic network (Section 6.1 of [RFC8345]) and network topology (Section 6.2 of [RFC8345]) data models with L2-specific topology attributes. An example is provided in Appendix B.

There are multiple applications for such a data model. For example, within the context of Interface to the Routing System (I2RS), nodes within the network can use the data model to capture their understanding of the overall network topology and expose it to a network controller. A network controller can then use the instantiated topology data to compare and reconcile its own view of the network topology with that of the network elements that it controls. Alternatively, nodes within the network may compare and reconcile this understanding either among themselves or with the help of a controller. Beyond the network element and the immediate context of I2RS itself, a network controller might even use the data model to represent its view of the topology that it controls and expose it to external applications. Further use cases where the data model can be applied are described in [I2RS-UR].

This document uses the common YANG types defined in [RFC6991] and adopts the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terminology for describing YANG modules is defined in [RFC7950]. The meanings of the symbols used in the tree diagram are defined in [RFC8340].

3. Layer 2 Topology Model

The Layer 2 network topology YANG module is designed to be generic and applicable to Layer 2 networks built with different Layer 2 technologies. It can be used to describe both the physical and the logical (virtual) Layer 2 network topologies.

The relationship between the Layer 2 topology module and the generic network and network topology module is shown in Figure 1. In order to represent a Layer 2 network topology, the generic network and topology models are augmented with Layer 2 specific information, such as the identifiers, identities (e.g., Provider Backbone Bridging [IEEE802.1ah], QinQ [IEEE802.1ad], or VXLAN [RFC7348]), attributes, and states of the Layer 2 networks, nodes, links, and termination points. Some of the information may be collected via Link Layer Discovery Protocol (LLDP) [IEEE802.1AB] or other Layer 2 protocols, and some of them may be locally configured.

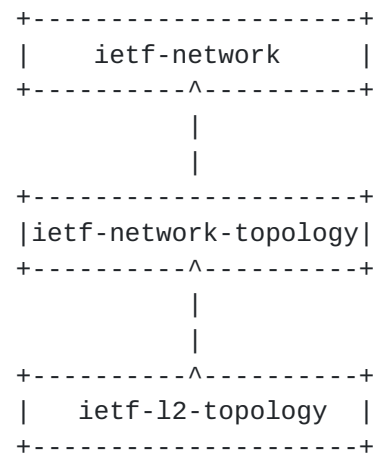

Figure 1: Layer 2 Topology YANG Module Structure

The structure of the "ietf-l2-topology" YANG module is depicted in the following tree diagram:

```
module: ietf-l2-topology
  augment /nw:networks/nw:network/nw:network-types:
    +--rw l2-network!
  augment /nw:networks/nw:network:
    +--rw l2-topology-attributes
       +--rw name?    string
       +--rw flags*   l2-flag-type
  augment /nw:networks/nw:network/nw:node:
    +--rw l2-node-attributes
       +--rw name?                 string
       +--rw description?          string
       +--rw management-address*   inet:ip-address
       +--rw sys-mac-address?      yang:mac-address
       +--rw management-vlan-id?   dot1q-types:vlanid {VLAN}?
       +--rw flags*                node-flag-type
  augment /nw:networks/nw:network/nt:link:
    +--rw l2-link-attributes
       +--rw name?    string
       +--rw flags*   link-flag-type
       +--rw rate?    uint64
       +--rw delay?   uint32
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw l2-termination-point-attributes
       +--rw description?                   string
       +--rw maximum-frame-size?            uint32
       +--rw (l2-termination-point-type)?
       |  +--:(ethernet)
       |  |  +--rw mac-address?             yang:mac-address
       |  |  +--rw eth-encapsulation?       identityref
```

```
       |  |  +--rw lag?                    boolean
       |  |  +--rw member-link-tp*         -> /nw:networks/network
                                           /node/nt:termination-point/tp-id
       |  |  +--rw auto-negotiation?       boolean
       |  |  +--rw duplex?                 duplex-mode
       |  |  +--rw default-untagged-vlan?  dot1q-types:vlanid {VLAN}?
       |  |  +--rw vlans* [vlan-id] {VLAN}?
       |  |  |  +--rw vlan-id    dot1q-types:vlanid
       |  |  |  +--rw name?      string
       |  |  +--rw qinq* [svlan-id cvlan-id] {QinQ}?
       |  |  |  +--rw svlan-id    dot1q-types:vlanid
       |  |  |  +--rw cvlan-id    dot1q-types:vlanid
       |  |  +--rw vxlan {VXLAN}?
       |  |     +--rw vni-id?   vni
       |  +--:(legacy)
       |     +--rw layer-2-address?        yang:phys-address
       |     +--rw encapsulation?          identityref
       +--ro tp-state?                     identityref

  notifications:
    +---n l2-node-event
    |  +--ro event-type?          l2-network-event-type
    |  +--ro node-ref?            -> /nw:networks/network
                                [nw:network-id=current()/../network-ref]/node/
node-id
    |  +--ro network-ref?         -> /nw:networks/network/network-id
    |  +--ro l2-network!
    |  +--ro l2-node-attributes
    |     +--ro name?                string
    |     +--ro description?         string
    |     +--ro management-address*  inet:ip-address
    |     +--ro sys-mac-address?     yang:mac-address
    |     +--ro management-vlan-id?  dot1q-types:vlanid {VLAN}?
    |     +--ro flags*               node-flag-type
    +---n l2-link-event
    |  +--ro event-type?          l2-network-event-type
    |  +--ro link-ref?            -> /nw:networks/network
                               [nw:network-id=current()/../network-ref]/nt:link/
link-id
    |  +--ro network-ref?         -> /nw:networks/network/network-id
    |  +--ro l2-network!
    |  +--ro l2-link-attributes
    |     +--ro name?    string
    |     +--ro flags*   link-flag-type
    |     +--ro rate?    uint64
    |     +--ro delay?   uint32
    +---n l2-termination-point-event
       +--ro event-type?                    l2-network-event-type
```

```
       +--ro tp-ref?                              -> /nw:networks/network
                                [nw:network-id=current()/../network-ref]/node
```

```
                          [nw:node-id=current()/../node-ref]/nt:termination-
point/tp-id
       +--ro node-ref?                          -> /nw:networks/network
                              [nw:network-id=current()/../network-ref]/node/
node-id
       +--ro network-ref?                       -> /nw:networks/network/
network-id
       +--ro l2-network!
       +--ro l2-termination-point-attributes
          +--ro description?                    string
          +--ro maximum-frame-size?             uint32
          +--ro (l2-termination-point-type)?
          |  +--:(ethernet)
          |  |  +--ro mac-address?              yang:mac-address
          |  |  +--ro eth-encapsulation?        identityref
          |  |  +--ro lag?                      boolean
          |  |  +--ro member-link-tp*           -> /nw:networks/network
                                                /node/nt:termination-point/tp-id
          |  |  +--ro auto-nego?                boolean
          |  |  +--ro duplex?                   duplex-mode
          |  |  +--ro default-untagged-vlan?    dot1q-types:vlanid {VLAN}?
          |  |  +--ro vlans* [vlan-id] {VLAN}?
          |  |  |  +--ro vlan-id    dot1q-types:vlanid
          |  |  |  +--ro name?      string
          |  |  +--ro qinq* [svlan-id cvlan-id] {QinQ}?
          |  |  |  +--ro svlan-id    dot1q-types:vlanid
          |  |  |  +--ro cvlan-id    dot1q-types:vlanid
          |  |  +--ro vxlan {VXLAN}?
          |  |     +--ro vni-id?   vni
          |  +--:(legacy)
          |     +--ro layer-2-address?          yang:phys-address
          |     +--ro encapsulation?            identityref
          +--ro tp-state?                       identityref
```

The Layer 2 topology YANG module augments the "ietf-network" and "ietf-network-topology" YANG modules as follows:

- A new network type "l2-network-type" is introduced. This is represented by a container object, and is inserted under the "network-types" container of the generic "ietf-network" module defined in Section 6.1 of [RFC8345].

- Additional network attributes are introduced in a grouping "l2-network-attributes", which augments the "network" list of the "ietf-network" module. The attributes include Layer 2 network name and a set of flags. Each type of flag is represented by a separate identity.

- Additional data objects for Layer 2 nodes are introduced by augmenting the "node" list of the generic "ietf-network" module.

New objects include Layer 2 node identifier, description, management address, and a set of flags.

- Additional data objects for Layer 2 termination points are introduced by augmenting the "termination-point" list of the "ietf-network-topology" module defined in Section 6.2 of [RFC8345]. New objects include Layer 2 termination point descriptions, Layer 2 termination point type specific attributes and Layer 2 termination point states.

- Links in the "ietf-network-topology" module are augmented as well with a set of Layer 2 parameters, allowing to associate a link with a name, a set of Layer 2 link attributes, and flags.

- Some optional Layer 2 technology specific attributes are introduced in this module as Layer 2 features because these attributes may be useful to expose to above services/applications. Note that learning or configuring advanced Layer 2 technology-specific attributes is not within the scope of the Layer 2 Topology YANG module; dedicated YANG modules should be used instead (e.g., [I-D.ietf-trill-yang]).

4. Layer 2 Topology YANG Module

This module uses types defined in [RFC6991], [RFC7224], [IEEE802.1Qcp], and [RFC8345]. It also references [RFC4761], [RFC4762], and [RFC4202].

```
<CODE BEGINS> file "ietf-l2-topology@2020-06-29.yang"
module ietf-l2-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2-topology";
  prefix l2t;

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991:Common YANG Data Types";
```

```
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991:Common YANG Data Types";
  }
  import iana-if-type {
    prefix ift;
    reference
      "RFC 7224: IANA Interface Type YANG Module";
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
    reference
      "IEEE Std 802.1Qcp-2018: Bridges and Bridged
       Networks - Amendment: YANG Data Model";
  }

  organization
    "IETF I2RS (Interface to the Routing System) Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/i2rs/>
     WG List:  <mailto:i2rs@ietf.org>

     Editor:    Jie Dong
                <mailto:jie.dong@huawei.com>

     Editor:    Xiugang Wei
                <mailto:weixiugang@huawei.com>

     Editor:    Qin Wu
                <mailto:bill.wu@huawei.com>

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

     Editor:   Anders Liu
                <andersliu@tencent.com>";
  description
    "This module defines a basic model for the Layer 2 topology
     of a network.

     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
     (http://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX; see
     the RFC itself for full legal notices.";

  revision 2020-06-29 {
    description
      "Initial revision";
    reference
      "RFC XXXX: A YANG Data Model for Layer 2
       Network Topologies";
  }

  feature VLAN {
    description
      "Enables VLAN tag support as defined in IEEE 802.1Q.";
    reference
      "IEEE Std 802.1Q-2014: Bridges and Bridged Networks";
  }

  feature QinQ {
    description
      "Enables QinQ double tag support as defined in IEEE 802.1ad.";
    reference
      "IEEE Std 802.1ad: Provider Bridges";
  }

  feature VXLAN {
    description
      "Enables VXLAN support as defined in RFC7348.";
    reference
      "RFC 7348: Virtual eXtensible Local Area  Network (VXLAN):
       A Framework for Overlaying Virtualized Layer 2
       Networks over Layer 3 Networks";
  }

  identity flag-identity {
    description
      "Base type for flags.";
  }

  identity eth-encapsulation-type {
    base ift:iana-interface-type;
    description
      "Base identity from which specific Ethernet
       encapsulation types are derived.";
```

```
    reference
      "RFC 7224: IANA Interface Type YANG Module";
  }

  identity ethernet {
    base eth-encapsulation-type;
    description
      "Native Ethernet encapsulation.";
  }

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

  identity qinq {
    base eth-encapsulation-type;
    description
      "QinQ encapsulation.";
  }

  identity pbb {
    base eth-encapsulation-type;
    description
      "Provider-backbone-bridging (PBB) encapsulation.
       The PBB functions are developed in IEEE 802.1ah.";
  }

  identity trill {
    base eth-encapsulation-type;
    description
      "TRILL encapsulation.";
  }

  identity vpls {
    base eth-encapsulation-type;
    description
      "Ethernet VPLS interface encapsulation.";
  }

  identity vxlan {
    base eth-encapsulation-type;
    description
      "VXLAN MAC in UDP encapsulation.";
    reference
      "RFC 7348: Virtual eXtensible Local Area  Network (VXLAN):
       A Framework for Overlaying Virtualized Layer 2
```

```
       Networks over Layer 3 Networks";
  }

  identity tp-state-type {
    description
      "Base type for termination point state.";
  }

  identity inuse {
    base tp-state-type;
    description
      "The termination point is in forwarding state.";
  }

  identity blocking {
    base tp-state-type;
    description
      "The termination point is in blocking state.";
  }

  identity down {
    base tp-state-type;
    description
      "The termination point is in down state.";
  }

  identity unknown {
    base tp-state-type;
    description
      "The termination point is in unknown state.";
  }

  typedef vni {
    type uint32 {
      range "0..16777215";
    }
    description
      "VXLAN Network Identifier or VXLAN Segment ID.
       It allows up to 16 M VXLAN segments to coexist
       within the same administrative domain.

       The use of value '0' is implementation-specific.";
    reference
      "RFC 7348: Virtual eXtensible Local Area  Network (VXLAN):
                 A Framework for Overlaying Virtualized Layer 2
                 Networks over Layer 3 Networks";
  }
```

```
  typedef l2-flag-type {
    type identityref {
      base flag-identity;
    }
    description
      "Base type for L2 flags. One example of L2 flag
       type is trill which represents trill topology
       type.";
  }

  typedef node-flag-type {
    type identityref {
      base flag-identity;
    }
    description
      "Node flag attributes. The physical node can be
       one example of node flag attribute.";
  }

  typedef link-flag-type {
    type identityref {
      base flag-identity;
    }
    description
      "Link flag attributes. One example of link flag
       attribute is the pseudowire.";
  }

  typedef l2-network-event-type {
    type enumeration {
      enum addition {
        value 0;
        description
          "A Layer 2 node or link or termination-point
           has been added.";
      }
      enum removal {
        value 1;
        description
          "A Layer 2 node or link or termination-point
           has been removed.";
      }
      enum update {
        value 2;
        description
          "A Layer 2 node or link or termination-point
           has been updated.";
      }
```

```
    }
    description
      "Layer 2 network event type for notifications.";
  }

  typedef duplex-mode {
    type enumeration {
      enum full-duplex {
        description
          "Indicates full-duplex mode.";
      }
      enum half-duplex {
        description
          "Indicates half-duplex mode.";
      }
    }
    description
      "Indicates the type of the duplex mode.";
  }

  grouping l2-network-type {
    description
      "Indicates the topology type to be L2.";
    container l2-network {
      presence "Indicates L2 Network";
      description
        "The presence of the container node indicates
         L2 Topology.";
    }
  }

  grouping l2-topology-attributes {
    description
      "L2 Topology scope attributes.";
    container l2-topology-attributes {
      description
        "Contains L2 topology attributes.";
      leaf name {
        type string;
        description
          "Name of the topology.";
      }
      leaf-list flags {
        type l2-flag-type;
        description
          "Topology flags.";
      }
    }
```

```
  }

  grouping l2-node-attributes {
    description
      "L2 node attributes";
    container l2-node-attributes {
      description
        "Contains L2 node attributes.";
      leaf name {
        type string;
        description
          "Node name.";
      }
      leaf description {
        type string;
        description
          "Node description.";
      }
      leaf-list management-address {
        type inet:ip-address;
        description
          "IP address used for
           management purpose.";
      }
      leaf sys-mac-address {
        type yang:mac-address;
        description
          "MAC address used to identify layer 2 node.
           This MAC address can be used as interface
           MAC address.";
      }
      leaf management-vlan-id {
        if-feature "VLAN";
        type dot1q-types:vlanid;
        description
          "VLAN ID used for management
           purpose.";
      }
      leaf-list flags {
        type node-flag-type;
        description
          "Node flags. It can be used to indicates
           node flag attributes.";
      }
    }
  }

  grouping l2-link-attributes {
```

```
    description
      "L2 link attributes";
    container l2-link-attributes {
      description
        "Contains L2 link attributes.";
      leaf name {
        type string;
        description
          "Link name.";
      }
      leaf-list flags {
        type link-flag-type;
        description
          "Link flags. It can be used to indicate
           link flag attributes.";
      }
      leaf rate {
        type uint64;
        units "Kbps";
        description
          "Link rate. It specifies bandwidth requirements
           associated with the specific link. The link
           contains a source and a destination.";
      }
      leaf delay {
        type uint32;
        units "microseconds";
        description
          "Unidirectional Link delay in
           microseconds.";
      }
    }
  }

  grouping l2-termination-point-attributes {
    description
      "L2 termination point attributes";
    container l2-termination-point-attributes {
      description
        "Containing L2 termination point attributes.";
      leaf description {
        type string;
        description
          "L2 termination point description. It also
           can be seen as port description.";
      }
      leaf maximum-frame-size {
        type uint32;
```

```
        description
          "Maximum L2 frame size that can be transported on
           the data link layer, e.g. Ethernet frame.
           If L2 frame is an Ethernet frame, the maximum frame size
           should include Frame Check Sequence(FCS) and should not
           include preamble,Start Frame Delimiter and Inter frame
           gap.";
      }
      choice l2-termination-point-type {
        description
          "Indicates termination-point type
           specific attributes.";
        case ethernet {
          leaf mac-address {
            type yang:mac-address;
            description
              "Interface MAC address.";
          }
          leaf eth-encapsulation {
            type identityref {
              base eth-encapsulation-type;
            }
            description
              "Encapsulation type of this
               termination point.";
          }
          leaf lag {
            type boolean;
            default "false";
            description
              "Defines whether lag is enabled or disabled.
               When it is set to true, the lag is enabled.";
          }
          leaf-list member-link-tp {
            when "../lag = 'true'" {
              description
                "Relevant only when the lag interface is enabled.";
            }
            type leafref {
              path "/nw:networks/nw:network/nw:node/nt:termination-point/nt:tp-
id";
            }
            description
              "Member link termination points.";
          }
          leaf auto-nego {
            type boolean;
            default "true";
```

description

```
            "Set to true if auto negotiation is supported.
             Set to false if auto negotiation is not supported.";
          }
          leaf duplex {
            type duplex-mode;
            description
              "Expose the duplex mode, full duplex or half-duplex.";
          }
          leaf default-untagged-vlan {
            when "derived-from-or-self(../eth-encapsulation, 'l2t:vlan')" {
              description
                "Only applies when the type of the Ethernet
                 encapsulation is 'vlan'.";
            }
            if-feature "VLAN";
            type dot1q-types:vlanid;
            description
              "Port VLAN ID is the VLAN identifier that
               will be assigned to any untagged frames entering
               the switch on the specific port.";
          }
          list vlans {
            when "derived-from-or-self(../eth-encapsulation, 'l2t:vlan')" {
              description
                "Only applies when the type of the Ethernet
                 encapsulation is 'vlan'.";
            }
            if-feature "VLAN";
            key "vlan-id";
            description
              "Interface configured VLANs.";
            leaf vlan-id {
              type dot1q-types:vlanid;
              description
                "VLAN ID.";
            }
            leaf name {
              type string {
                length "1..31";
              }
              description
                "VLAN name.";
            }
          }
          list qinq {
            when "derived-from-or-self(../eth-encapsulation, 'l2t:qinq')" {
              description
                "Only applies when the type of the Ethernet
```

```
             encapsulation is 'qinq'.";
        }
        if-feature "QinQ";
        key "svlan-id cvlan-id";
        description
          "Interface configured SVLANs and CVLANs.";
        leaf svlan-id {
          type dot1q-types:vlanid;
          description
            "Service VLAN ID.";
        }
        leaf cvlan-id {
          type dot1q-types:vlanid;
          description
            "Customer VLAN ID.";
        }
      }
      container vxlan {
        when "derived-from-or-self(../eth-encapsulation, 'l2t:vxlan')" {
          description
            "Only applies when the type of the Ethernet
             encapsulation is 'vxlan'.";
        }
        if-feature "VXLAN";
        leaf vni-id {
          type vni;
          description
            "VXLAN Network Identifier (VNI).";
        }
        description
          "Vxlan encapsulation type.";
      }
    }
    case legacy {
      leaf layer-2-address {
        type yang:phys-address;
        description
          "Interface Layer 2 address.";
      }
      leaf encapsulation {
        type identityref {
          base ift:iana-interface-type;
        }
        description
          "Other legacy encapsulation type of this
           termination point.";
      }
    }
```

```
      }
      leaf tp-state {
        type identityref {
          base tp-state-type;
        }
        config false;
        description
          "State of the termination point.";
      }
    }
  }

  augment "/nw:networks/nw:network/nw:network-types" {
    description
      "Introduces new network type for L2 topology.";
    uses l2-network-type;
  }
  augment "/nw:networks/nw:network" {
    when '/nw:networks/nw:network/nw:network-types/l2t:l2-network' {
      description
        "Augmentation parameters apply only for networks
         with L2 topology.";
    }
    description
      "Configuration parameters for the L2 network
       as a whole.";
    uses l2-topology-attributes;
  }
  augment "/nw:networks/nw:network/nw:node" {
    when '/nw:networks/nw:network/nw:network-types/l2t:l2-network' {
      description
        "Augmentation parameters apply only for networks
         with L2 topology.";
    }
    description
      "Configuration parameters for L2 at the node
       level.";
    uses l2-node-attributes;
  }
  augment "/nw:networks/nw:network/nt:link" {
    when '/nw:networks/nw:network/nw:network-types/l2t:l2-network' {
      description
        "Augmentation parameters apply only for networks
         with L2 topology.";
    }
    description
      "Augments L2 topology link information.";
    uses l2-link-attributes;
```

```
  }
  augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
    when '/nw:networks/nw:network/nw:network-types/l2t:l2-network' {
      description
        "Augmentation parameters apply only for networks
         with L2 topology.";
    }
    description
      "Augments L2 topology termination point information.";
    uses l2-termination-point-attributes;
  }

  notification l2-node-event {
    description
      "Notification event for L2 node.";
    leaf event-type {
      type l2-network-event-type;
      description
        "Event type.";
    }
    uses nw:node-ref;
    uses l2-network-type;
    uses l2-node-attributes;
  }

  notification l2-link-event {
    description
      "Notification event for L2 link.";
    leaf event-type {
      type l2-network-event-type;
      description
        "Event type.";
    }
    uses nt:link-ref;
    uses l2-network-type;
    uses l2-link-attributes;
  }

  notification l2-termination-point-event {
    description
      "Notification event for L2 termination point.";
    leaf event-type {
      type l2-network-event-type;
      description
        "Event type.";
    }
    uses nt:tp-ref;
    uses l2-network-type;
```

```
    uses l2-termination-point-attributes;
  }
}
```

<CODE ENDS>

5. IANA Considerations

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

```
URI: urn:ietf:params:xml:ns:yang:ietf-l2-topology
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

URI: urn:ietf:params:xml:ns:yang:ietf-l2-topology-state
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

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

```
name: ietf-l2-topology
namespace: urn:ietf:params:xml:ns:yang:ietf-l2-topology
prefix: l2t
reference: RFC XXXX

name: ietf-l2-topology-state
namespace: urn:ietf:params:xml:ns:yang:ietf-l2-topology-state
prefix: l2t-s
reference: RFC XXXX
```

These modules are not maintained by IANA.

6. Security Considerations

The YANG module specified in this document defines a schema 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 Layer 2 topology module defines information that can be configurable in certain instances, for example in the case of virtual topologies that can be created by client applications. In such cases, a malicious client could introduce topologies that are undesired. Specifically, a malicious client could attempt to remove or add a node, a link, a termination point, by creating or deleting corresponding elements in the node, link, and termination point lists, respectively. In the case of a topology that is learned, the server will automatically prohibit such misconfiguration attempts. In the case of a topology that is configured, i.e. whose origin is "intended", the undesired configuration could become effective and be reflected in the operational state datastore [RFC8342], leading to disruption of services provided via this topology. For those reasons, it is important that the NACM is vigorously applied to prevent topology misconfiguration by unauthorized clients.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- l2-network-attributes: A malicious client could attempt to sabotage the configuration of any of the contained attributes, such as the name or the flag data nodes.

- l2-node-attributes: A malicious client could attempt to sabotage the configuration of important node attributes, such as the name or the management-address.

- l2-link-attributes: A malicious client could attempt to sabotage the configuration of important link attributes, such as the rate or the delay data nodes.

- l2-termination-point-attributes: A malicious client could attempt to sabotage the configuration of important termination point attributes (e.g., 'maximum-frame-size').

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. In particular, the YANG model for layer 2 topology may expose sensitive information, for example the

MAC addresses of devices, VLAN/VXLAN identifiers. Unrestricted use of such information can lead to privacy violations. For example, listing MAC addresses in a network allows monitoring of devices and their movements. Location information can be derived from MAC addresses of network devices, bypassing protection of location information by the Operating System.

7. Acknowledgements

The authors would like to acknowledge the comments and suggestions received from Susan Hares, Alia Atlas, Juergen Schoenwaelder, Mach Chen, Alexander Clemm, Sriganesh Kini, Oscar Gonzalez de Dios, Stig Venaas, Christian Huitema, and Meral Shirazipour,Benjamin Kaduk.

Many thanks to Ladislav Lhotka for the yang-doctors review.

8. References

8.1. Normative References

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

[RFC4202] Kompella, K., Ed. and Y. Rekhter, Ed., "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, DOI 10.17487/RFC4202, October 2005, <https://www.rfc-editor.org/info/rfc4202>.

[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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc6020>.

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

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <https://www.rfc-editor.org/info/rfc7224>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

[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-trill-yang]
Weiguo, H., Yizhou, L., Kumar, D., Durrani, M., Zhai, H., and L. Xia, "TRILL YANG Data Model", draft-ietf-trill-yang-04 (work in progress), December 2015.

[I2RS-UR] Hares, S. and M. Chen, "Summary of I2RS Use Case Requirements", draft-ietf-i2rs-usecase-reqs-summary-03 (work in progress), November 2016.

[IEEE802.1AB]
"Station and Media Access Control Connectivity Discovery", IEEE Std 802.1AB-2016, March 2016.

[IEEE802.1ad]
"Virtual Bridged Local Area Networks Amendment 4: Provider Bridges", IEEE Std 802.1ad-2005, May 2006.

[IEEE802.1ah]
"Virtual Bridged Local Area Networks Amendment 4: Provider Bridges", IEEE Std 802.1ah-2008, August 2008.

[IEEE802.1Qcp]
"Bridges and Bridged Networks - Amendment: YANG Data Model", IEEE Std 802.1Qcp-2018, September 2018.

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

[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>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[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>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

Appendix A. Companion YANG Module for Non-NMDA Compliant Implementations

The YANG module ietf-l2-topology defined in this document augments two modules, "ietf-network" and "ietf-network-topology", that are designed to be used in conjunction with implementations that support

the Network Management Datastore Architecture (NMDA) defined in [RFC8342]. In order to allow implementations to use the model even in cases when NMDA is not supported, a set of companion modules have been defined that represent a state model of networks and network topologies, "ietf-network-state" and "ietf-network-topology-state", respectively.

In order to be able to use the model for layer 2 topologies defined in this document in conjunction with non-NMDA compliant implementations, a corresponding companion module is defined that represents the operational state of layer 2 network topologies. The module "ietf-l2-topology-state" mirrors the module "ietf-l2-topology" defined in Section 4. However, it augments "ietf-network-state" and "ietf-network-topology-state" (instead of "ietf-network" and "ietf-network-topology") and all its data nodes are non-configurable.

The companion module "ietf-l2-topology" SHOULD NOT be supported by implementations that support NMDA. It is for this reason that this module is defined in the informative Appendix.

As the structure of this modules mirrors that of its underlying modules, the YANG tree is not depicted separately.

```
<CODE BEGINS> file "ietf-l2-topology-state@2020-06-29.yang"
module ietf-l2-topology-state {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2-topology-state";
  prefix l2t-s;

  import ietf-network-state {
    prefix nw-s;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-network-topology-state {
    prefix nt-s;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }
  import ietf-l2-topology {
    prefix l2t;
    reference
      "RFC XXXX: A YANG Data Model for Layer 2 Network
                 Topologies";
  }

  organization
    "IETF I2RS (Interface to the Routing System) Working Group";
```

```
    contact
      "WG Web:   <http://tools.ietf.org/wg/i2rs/>
       WG List:  <mailto:i2rs@ietf.org>

       Editor:    Jie Dong
                  <mailto:jie.dong@huawei.com>
       Editor:    Xiugang Wei
                  <mailto:weixiugang@huawei.com>
       Editor:    Qin Wu
                  <mailto:bill.wu@huawei.com>
       Editor:    Mohamed Boucadair
                  <mailto:mohamed.boucadair@orange.com>
       Editor:   Anders Liu
                  <andersliu@tencent.com>";
    description
      "This module defines a model for Layer 2 Network Topology
       state, representing topology that either is learned or
       results from applying topology that has been configured per
       the 'ietf-l2-topology' model, mirroring the
       corresponding data nodes in this model.

       This model mirrors 'ietf-l2-topology' but contains only
       read-only state data.  The model is not needed when the
       underlying implementation infrastructure supports the
       Network Management Datastore Architecture (NMDA).

       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
       (http://trustee.ietf.org/license-info).

       This version of this YANG module is part of RFC XXXX; see
       the RFC itself for full legal notices.";

    revision 2020-06-29 {
      description
        "Initial revision";
      reference
        "RFC XXXX: A YANG Data Model for Layer 2 Network
                   Topologies";
    }

    /*
```

```
     * Data nodes
     */

    augment "/nw-s:networks/nw-s:network/nw-s:network-types" {
      description
        "Introduces a new network type for L2 topology.";
      uses l2t:l2-network-type;
    }

    augment "/nw-s:networks/nw-s:network" {
      when '/nw-s:networks/nw-s:network/nw-s:network-types/'
         + 'l2t-s:l2-network' {
        description
          "Augmentation parameters apply only for networks
           with L2 topology.";
      }
      description
        "Configuration parameters for the L2 network
         as a whole.";
      uses l2t:l2-network-attributes;
    }

    augment "/nw-s:networks/nw-s:network/nw-s:node" {
      when '../nw-s:network-types/l2t-s:l2-network' {
        description
          "Augmentation parameters apply only for networks
           with L2 topology.";
      }
      description
        "Configuration parameters for L2 at the node
         level.";
      uses l2t:l2-node-attributes;
    }

    augment "/nw-s:networks/nw-s:network/nt-s:link" {
      when '../nw-s:network-types/l2t-s:l2-network' {
        description
          "Augmentation parameters apply only for networks
           with L2 topology.";
      }
      description
        "Augments L2 topology link information.";
      uses l2t:l2-link-attributes;
    }

    augment "/nw-s:networks/nw-s:network/nw-s:node/"
          + "nt-s:termination-point" {
      when '../../nw-s:network-types/l2t-s:l2-network' {
```

```
        description
          "Augmentation parameters apply only for networks
           with L2 topology.";
      }
      description
        "Augments L2 topology termination point information.";
      uses l2t:l2-termination-point-attributes;
    }

    /*
     * Notifications
     */

    notification l2-node-event {
      description
        "Notification event for L2 node.";
      leaf event-type {
        type l2t:l2-network-event-type;
        description
          "Event type.";
      }
      uses nw-s:node-ref;
      uses l2t:l2-network-type;
      uses l2t:l2-node-attributes;
    }

    notification l2-link-event {
      description
        "Notification event for a L2 link.";
      leaf event-type {
        type l2t:l2-network-event-type;
        description
          "Event type.";
      }
      uses nt-s:link-ref;
      uses l2t:l2-network-type;
      uses l2t:l2-link-attributes;
    }

    notification l2-termination-point-event {
      description
        "Notification event for L2 termination point.";
      leaf event-type {
        type l2t:l2-network-event-type;
        description
          "Event type.";
      }
      uses nt-s:tp-ref;
```

```
      uses l2t:l2-network-type;
      uses l2t:l2-termination-point-attributes;
    }
  }
  <CODE ENDS>
```

Appendix B. An Example

This section contains an example of an instance data tree in JSON encoding [RFC7951]. The example instantiates "ietf-l2-topology" for the topology that is depicted in the following diagram. There are three nodes: D1, D2, and D3. D1 has three termination points: 1-0-1, 1-2-1, and 1-3-1. D2 has three termination points as well: 2-1-1, 2-0-1, and 2-3-1. D3 has two termination points: 3-1-1 and 3-2-1. In addition, there are six links, two between each pair of nodes, with one going in each direction.

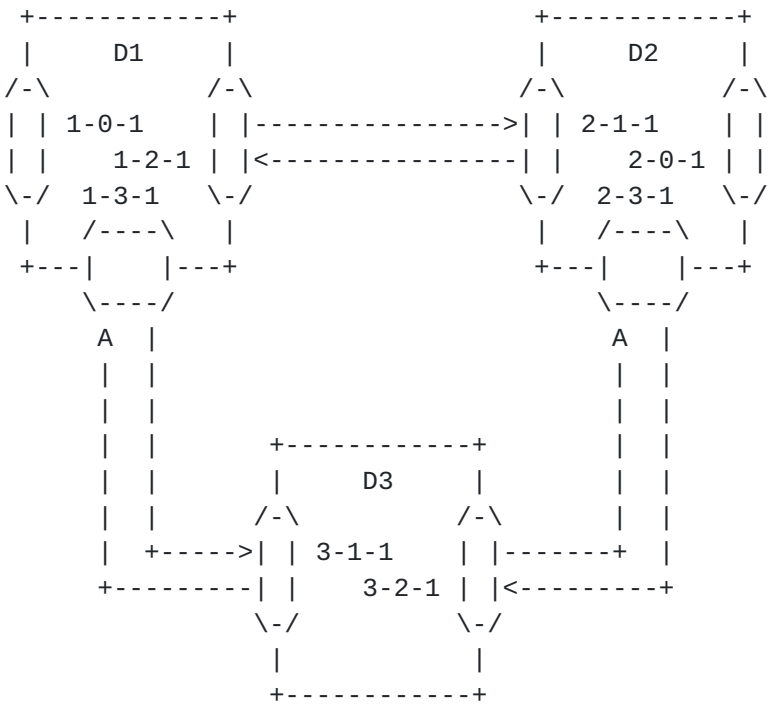

Figure 2. A Network Topology Example

The corresponding instance data tree is depicted below:

```
 {
  "ietf-network:networks": {
    "network": [
      {
        "network-types": {
          "ietf-l2-topology:l2-network": {}
```

```
          },
          "network-id": "l2-topo-example",
          "node": [
            {
              "node-id": "D1",
              "ietf-network-topology:termination-point": [
                {
                  "tp-id": "1-0-1",
                  "ietf-l2-topology:l2-termination-point-attributes": {
                    "mac-address": "00:00:5E:00:53:D0"
                  }
                },
                {
                  "tp-id": "1-2-1",
                  "ietf-l2-topology:l2-termination-point-attributes": {
                    "mac-address": "00:00:5E:00:53:D1"
                  }
                },
                {
                  "tp-id": "1-3-1",
                  "ietf-l2-topology:l2-termination-point-attributes": {
                    "mac-address": "00:00:5E:00:53:D2"
                  }
                }
              ],
              "ietf-l2-topology:l2-node-attributes": {
                "management-address": [
                  "192.0.2.1",
                  "2001:db8:0:1::/64"
                ]
              }
            },
            {
              "node-id": "D2",
              "ietf-network-topology:termination-point": [
                {
                  "tp-id": "2-0-1",
                  "ietf-l2-topology:l2-termination-point-attributes": {
                    "mac-address": "00:00:5E:00:53:E0"
                  }
                },
                {
                  "tp-id": "2-1-1",
                  "ietf-l2-topology:l2-termination-point-attributes": {
                    "mac-address": "00:00:5E:00:53:E1"
                  }
                },
                {
```

```
                "tp-id": "2-3-1",
                "ietf-l2-topology:l2-termination-point-attributes": {
                  "mac-address": "00:00:5E:00:53:E2"
                }
              }
            ],
            "ietf-l2-topology:l2-node-attributes": {
              "management-address": [
                "192.0.2.2",
                "2001:db8:0:2::/64"
              ]
            }
          },
          {
            "node-id": "D3",
            "ietf-network-topology:termination-point": [
              {
                "tp-id": "3-1-1",
                "ietf-l2-topology:l2-termination-point-attributes": {
                  "mac-address": "00:00:5E:00:53:F0"
                }
              },
              {
                "tp-id": "3-2-1",
                "ietf-l2-topology:l2-termination-point-attributes": {
                  "mac-address": "00:00:5E:00:53:F1"
                }
              }
            ],
            "ietf-l2-topology:l2-node-attributes": {
              "management-address": [
                "192.0.2.3",
               "2001:db8:0:3::/64"
              ]
            }
          }
        ],
        "ietf-network-topology:link": [
          {
            "link-id": "D1,1-2-1,D2,2-1-1",
            "source": {
              "source-node": "D1",
              "source-tp": "1-2-1"
            },
            "destination": {
              "dest-node": "D2",
              "dest-tp": "2-1-1"
            },
```

```
        "ietf-l2-topology:l2-link-attributes": {
          "rate": "1000"
        }
      },
      {
        "link-id": "D2,2-1-1,D1,1-2-1",
        "source": {
          "source-node": "D2",
          "source-tp": "2-1-1"
        },
        "destination": {
          "dest-node": "D1",
          "dest-tp": "1-2-1"
        },
        "ietf-l2-topology:l2-link-attributes": {
          "rate": "1000"
        }
      },
      {
        "link-id": "D1,1-3-1,D3,3-1-1",
        "source": {
          "source-node": "D1",
          "source-tp": "1-3-1"
        },
        "destination": {
          "dest-node": "D3",
          "dest-tp": "3-1-1"
        },
        "ietf-l2-topology:l2-link-attributes": {
          "rate": "1000"
        }
      },
      {
        "link-id": "D3,3-1-1,D1,1-3-1",
        "source": {
          "source-node": "D3",
          "source-tp": "3-1-1"
        },
        "destination": {
          "dest-node": "D1",
          "dest-tp": "1-3-1"
        },
        "ietf-l2-topology:l2-link-attributes": {
          "rate": "1000"
        }
      },
      {
        "link-id": "D2,2-3-1,D3,3-2-1",
```

```
              "source": {
                "source-node": "D2",
                "source-tp": "2-3-1"
              },
              "destination": {
                "dest-node": "D3",
                "dest-tp": "3-2-1"
              },
              "ietf-l2-topology:l2-link-attributes": {
                "rate": "1000"
              }
            },
            {
              "link-id": "D3,3-2-1,D2,2-3-1",
              "source": {
                "source-node": "D3",
                "source-tp": "3-2-1"
              },
              "destination": {
                "dest-node": "D2",
                "dest-tp": "2-3-1"
              },
              "ietf-l2-topology:l2-link-attributes": {
                "rate": "1000"
              }
            }
          ]
        }
      ]
    }
  }
```


Authors' Addresses

Jie Dong
Huawei
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: jie.dong@huawei.com

Xiugang Wei
Huawei
Huawei Campus, No. 156 Beiqing Rd.
Beijing 100095
China

Email: weixiugang@huawei.com

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

Email: bill.wu@huawei.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Anders Liu
Tecent
Yinke Building 38 Haidian St, Haidian District
Beijing 100080
China

Email: andersliu@tencent.com