A YANG Data Model for Layer 2 Network Topologies

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 topology attributes
   that are specific to Layer 2.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc8944.

Copyright Notice

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 L2-specific information, such as
the identifiers, identities (e.g., Provider Backbone Bridging
[IEEE802.1ah], QinQ [IEEE802.1ad], or Virtual eXtensible Local Area
Network (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.

```
                    +--------------------+
                    |    ietf-network    |
                    +----------^---------+
                               |
                               |
                    +--------------------+
                    |ietf-network-topology|
                    +----------^---------+
                               |
                               |
                    +----------^---------+
                    |   ietf-l2-topology |
                    +--------------------+
```
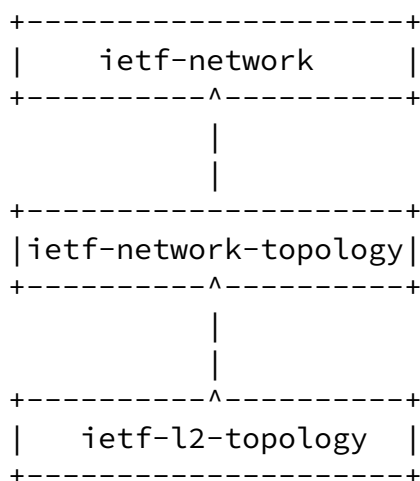
           Figure 1: Layer 2 Topology YANG Module Structure

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-topology!
```

```
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 flags*                node-flag-type
     +--rw bridge-id*            string
     +--rw management-address*   inet:ip-address
     +--rw management-mac?       yang:mac-address
     +--rw management-vlan?      string
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
     +--rw auto-nego?   boolean
     +--rw duplex?      duplex-mode
augment /nw:networks/nw:network/nw:node/nt:termination-point:
  +--rw l2-termination-point-attributes
     +--rw interface-name?     string
     +--rw mac-address?        yang:mac-address
     +--rw port-number*        uint32
     +--rw unnumbered-id*      uint32
     +--rw encapsulation-type? identityref
     +--rw outer-tag?          dot1q-types:vid-range-type {VLAN}?
     +--rw outer-tpid?         dot1q-types:dot1q-tag-type {QinQ}?
     +--rw inner-tag?          dot1q-types:vid-range-type {VLAN}?
     +--rw inner-tpid?         dot1q-types:dot1q-tag-type {QinQ}?
     +--rw lag?                boolean
     +--rw member-link-tp*
           -> /nw:networks/network/node/nt:termination-point/tp-id
     +--rw vxlan {VXLAN}?
        +--rw vni-id?   vni

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-topology!
  |  +--ro l2-node-attributes
  |     +--ro name?                 string
  |     +--ro flags*                node-flag-type
  |     +--ro bridge-id*            uint64
  |     +--ro management-address*   inet:ip-address
  |     +--ro management-mac?       yang:mac-address
```

```
|      +--ro management-vlan?       string
+---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-topology!
|   +--ro l2-link-attributes
|      +--ro name?        string
|      +--ro flags*       link-flag-type
|      +--ro rate?        uint64
|      +--ro delay?       uint32
|      +--ro auto-nego?   boolean
|      +--ro duplex?      duplex-mode
+---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-topology!
    +--ro l2-termination-point-attributes
       +--ro interface-name?     string
       +--ro mac-address?        yang:mac-address
       +--ro port-number*        uint32
       +--ro unnumbered-id*      uint32
       +--ro encapsulation-type?   identityref
       +--ro outer-tag?          dot1q-types:vid-range-type {VLAN}?
       +--ro outer-tpid?         dot1q-types:dot1q-tag-type {QinQ}?
       +--ro inner-tag?          dot1q-types:vid-range-type {VLAN}?
       +--ro inner-tpid?         dot1q-types:dot1q-tag-type {QinQ}?
       +--ro lag?                boolean
       +--ro member-link-tp*
          -> /nw:networks/network/node/nt:termination-point/tp-id
       +--ro vxlan {VXLAN}?
          +--ro vni-id?   vni
```

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 the 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 the Layer 2 node identifier, management
         address, management MAC address, management VLAN, 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 the interface name, encapsulation
         type, lag support indication, and attributes that are specific to
         the Layer 2 termination point type.

    *   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., [TRILL-YANG]).

4.   Layer 2 Topology YANG Module

     This module uses types defined in [RFC6991], [RFC7224],
     [IEEE802.1Qcp], and [RFC8345]. It also references [IEEE802.1Q-2014],
     [IEEE802.1ad], [RFC7348], and [RFC7727].

     <CODE BEGINS> file "ietf-l2-topology@2020-11-15.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 ianaift;
      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:    <https://datatracker.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
                  <mailto: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
```

  revision 2020-11-15 {
    description
      "Initial revision.";
    reference
      "RFC 8944: 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 RFC 7348.";
    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 ianaift: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
          "Transparent Interconnection of Lots of Links (TRILL)
           encapsulation.";
      }

      identity vpls {
        base eth-encapsulation-type;
        description
          "Ethernet Virtual Private LAN Service (VPLS)
           interface encapsulation.";
      }

      identity vxlan {
        base eth-encapsulation-type;
        description
          "VXLAN Media Access Control (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";
      }
```

```
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 the trill topology
     type.";
}

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

typedef link-flag-type {
  type identityref {
    base flag-identity;
  }
  description
    "Link flag attributes.  One example of a 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-topology {
      presence "Indicates L2 Network Topology.";
      description
        "The presence of the container node indicates
         L2 Network 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-list flags {
        type node-flag-type;
        description
          "Node flags.  It can be used to indicate
           node flag attributes.";
      }
      leaf-list bridge-id {
        type string {
          pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){7}';
        }
        description
          "This is the bridge identifier represented as a
           hexadecimal 8-octet string.  It has 4 bits of
           priority, 12 bits of Multiple Spanning Tree
           Instance Identifier (MSTI-ID), and the base bridge
           identifier.  There may be multiple for each
           spanning tree instance.";
        reference
          "RFC 7727: Spanning Tree Protocol (STP) Application of
                     the Inter-Chassis Communication Protocol
                     (ICCP)";
      }
      leaf-list management-address {
        type inet:ip-address;
        description
          "IP address used for management purpose.";
      }
      leaf management-mac {
```

```
            type yang:mac-address;
            description
              "This is a MAC address used for the bridge management.
               It can be the Bridge Base VLAN ID (VID), interface
               MAC address, or other. ";
          }
          leaf management-vlan {
            type string;
            description
              "This is a VLAN that supports the management address.
               The actual VLAN ID type and value would be a member of
               this VLAN.";
          }
        }
      }

      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.";
          }
          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
              "Exposes the duplex mode, full-duplex or half-duplex.";
          }
        }
      }

      grouping l2-termination-point-attributes {
        description
          "L2 termination point attributes.";
        container l2-termination-point-attributes {
          description
            "Containing L2 termination point attributes.";
          leaf interface-name {
            type string;
            description
              "Name of the interface.  The name can (but does not
               have to) correspond to an interface reference of a
               containing node's interface, i.e., the path name of a
               corresponding interface data node on the containing
               node is reminiscent of data type interface-ref defined
               in RFC 8343.  It should be noted that data type
               interface-ref of RFC 8343 cannot be used directly,
               as this data type is used to reference an interface
               in a datastore of a single node in the network, not
               to uniquely reference interfaces across a network.";
          }
          leaf mac-address {
            type yang:mac-address;
            description
              "Interface MAC address for logical link control.";
          }
          leaf-list port-number {
            type uint32;
            description
              " List of port numbers of the bridge ports for which each
                entry contains bridge management information.";
          }
          leaf-list unnumbered-id {
            type uint32;
            description
              "List of unnumbered interface identifiers.
               The unnumbered interface identifier will correspond to
               the ifIndex value of the interface, i.e., the ifIndex
               value of the ifEntry that represents the interface in
               implementations where the Interfaces Group MIB
```

```
                (RFC 2863) is supported.";
        }
        leaf encapsulation-type {
          type identityref {
            base eth-encapsulation-type;
          }
          description
            "Encapsulation type of this
                  termination point.";
        }
        leaf outer-tag {
          if-feature "VLAN";
          type dot1q-types:vid-range-type;
          description
            "The outermost VLAN tag.  It may include a list of VLAN
             Ids or nonoverlapping VLAN ranges.";
        }
        leaf outer-tpid {
          if-feature "QinQ";
          type dot1q-types:dot1q-tag-type;
          description
            "Identifies a specific 802.1Q tag type of outermost VLAN
             tag.";
        }
        leaf inner-tag {
          if-feature "VLAN";
          type dot1q-types:vid-range-type;
          description
            "The inner VLAN tag.  It may include a list of VLAN
             Ids or nonoverlapping VLAN ranges.";
        }
        leaf inner-tpid {
          if-feature "QinQ";
          type dot1q-types:dot1q-tag-type;
          description
            "Identifies a specific 802.1Q tag type of inner VLAN tag.";
        }
        leaf lag {
          type boolean;
          default "false";
          description
            "Defines whether lag is supported or not.
             When it is set to true, the lag is supported.";
        }
        leaf-list member-link-tp {
          when "../lag = 'true'" {
            description
              "Relevant only when the lag interface is supported.";
          }
          type leafref {
            path "/nw:networks/nw:network/nw:node"
```

```
              + "/nt:termination-point/nt:tp-id";
        }
        description
          "List of member link termination points associated with
           specific L2 termination point.";
    }
    container vxlan {
      when "derived-from-or-self(../encapsulation-type, "
        + "'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.";
    }
  }
}

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-topology' {
    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-topology' {
    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-topology' {
        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-topology' {
        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

    IANA has registered 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.

    IANA has registered 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 8944

    Name: ietf-l2-topology-state
    Namespace: urn:ietf:params:xml:ns:yang:ietf-l2-topology-state
    Prefix: l2t-s
    Reference: RFC 8944

    These modules are not maintained by IANA.

6. Security Considerations

    The YANG modules specified in this document define a schema for data
    that is designed to be accessed via network management protocols,
    such as Network Configuration Protocol (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, or 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 module
for Layer 2 topology may expose sensitive information, for example,
the MAC addresses of devices or 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. References

7.1. 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>.

   [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>.

   [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>.

   [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>.

   [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>.

7.2. Informative References

   [I2RS-UR] Hares, S. and M. Chen, "Summary of I2RS Use Case
              Requirements", Work in Progress, Internet-Draft, draft-
              ietf-i2rs-usecase-reqs-summary-03, 15 November 2016,
              <https://tools.ietf.org/html/draft-ietf-i2rs-usecase-reqs-
              summary-03>.

   [IEEE802.1AB]
              IEEE, "IEEE Standard for Local and metropolitan area
              networks - Station and Media Access Control Connectivity
              Discovery", IEEE Std 802.1AB-2016,
              DOI 10.1109/IEEESTD.2016.7433915, March 2016,
              <https://doi.org/10.1109/IEEESTD.2016.7433915>.

   [IEEE802.1ad]
              IEEE, "IEEE Standard for Local and Metropolitan Area
              Networks--Virtual Bridged Local Area Networks--Amendment
              4: Provider Bridges", IEEE Std 802.1ad-2005,
              DOI 10.1109/IEEESTD.2006.6044678, May 2006,
              <https://doi.org/10.1109/IEEESTD.2006.6044678>.

   [IEEE802.1ah]
              IEEE, "IEEE Standard for Local and metropolitan area
              networks -- Virtual Bridged Local Area Networks Amendment

                7: Provider Backbone Bridges", IEEE Std 802.1ah-2008,
                DOI 10.1109/IEEESTD.2008.4602826, August 2008,
                <https://doi.org/10.1109/IEEESTD.2008.4602826>.

     [IEEE802.1Q-2014]
                IEEE, "IEEE Standard for Local and metropolitan area
                networks--Bridges and Bridged Networks", IEEE 802.1Q-2014,
                DOI 10.1109/IEEESTD.2014.6991462, December 2014,
                <https://doi.org/10.1109/IEEESTD.2014.6991462>.

     [IEEE802.1Qcp]
                IEEE, "IEEE Standard for Local and metropolitan area
                networks--Bridges and Bridged Networks--Amendment 30: YANG
                Data Model", IEEE Std 802.1Qcp-2018,
                DOI 10.1109/IEEESTD.2018.8467507, September 2018,
                <https://doi.org/10.1109/IEEESTD.2018.8467507>.

     [RFC7727] Zhang, M., Wen, H., and J. Hu, "Spanning Tree Protocol
                (STP) Application of the Inter-Chassis Communication
                Protocol (ICCP)", RFC 7727, DOI 10.17487/RFC7727, January
                2016, <https://www.rfc-editor.org/info/rfc7727>.

     [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>.

     [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>.

     [TRILL-YANG]
                Hao, W., Li, Y., Kumar, D., Durrani, M., Zhai, H., and L.
                Xia, "TRILL YANG Data Model", Work in Progress, Internet-
                Draft, draft-ietf-trill-yang-04, 20 December 2015,
                <https://tools.ietf.org/html/draft-ietf-trill-yang-04>.

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 nonconfigurable.

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 module mirrors that of its underlying
modules, the YANG tree is not depicted separately.

```
<CODE BEGINS> file "ietf-l2-topology-state@2020-11-15.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 8944: 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 8944; see
     the RFC itself for full legal notices.";

  revision 2020-11-15 {
    description
      "Initial revision.";
    reference
      "RFC 8944: 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:network-types/l2t-s:l2-topology' {
      description
```

```
          "Augmentation parameters apply only for networks
           with L2 topology.";
    }
    description
      "Configuration parameters for the L2 network
       as a whole.";
    uses l2t:l2-topology-attributes;
}

augment "/nw-s:networks/nw-s:network/nw-s:node" {
    when '../nw-s:network-types/l2t-s:l2-topology' {
      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-topology' {
      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-topology' {
      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 an 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.
For termination point 1-0-1, it provides lag support and has two
member link termination points: 1-0-1-1 and 1-0-1-2. In addition,
there are six links, two between each pair of nodes with one going in
each direction.

```
                      +-----------+                    +-----------+
                      |    D1     |                    |    D2     |
         1-0-1-1 /-\            /-\                 /-\            /-\
        <--------->| | 1-0-1   | |----------------->| | 2-1-1   | |
         1-0-1-2 | |     1-2-1 | |<-----------------| |    2-0-1 | |
        <--------> \-/  1-3-1   \-/                 \-/  2-3-1    \-/
                    |   /----\   |                   |   /----\    |
                    +---|    |---+                   +---|    |---+
                        \----/                           \----/
                         A  |                             A  |
                         |  |                             |  |
                         |  |                             |  |
                         |  |     +-----------+           |  |
                         |  |     |    D3     |           |  |
                         |  |    /-\         /-\          |  |
                         | +----->| | 3-1-1 | |------+   |  |
                         +--------| |   3-2-1 | |<---------+
                                  \-/         \-/
                                   |           |
                                  +-----------+
```

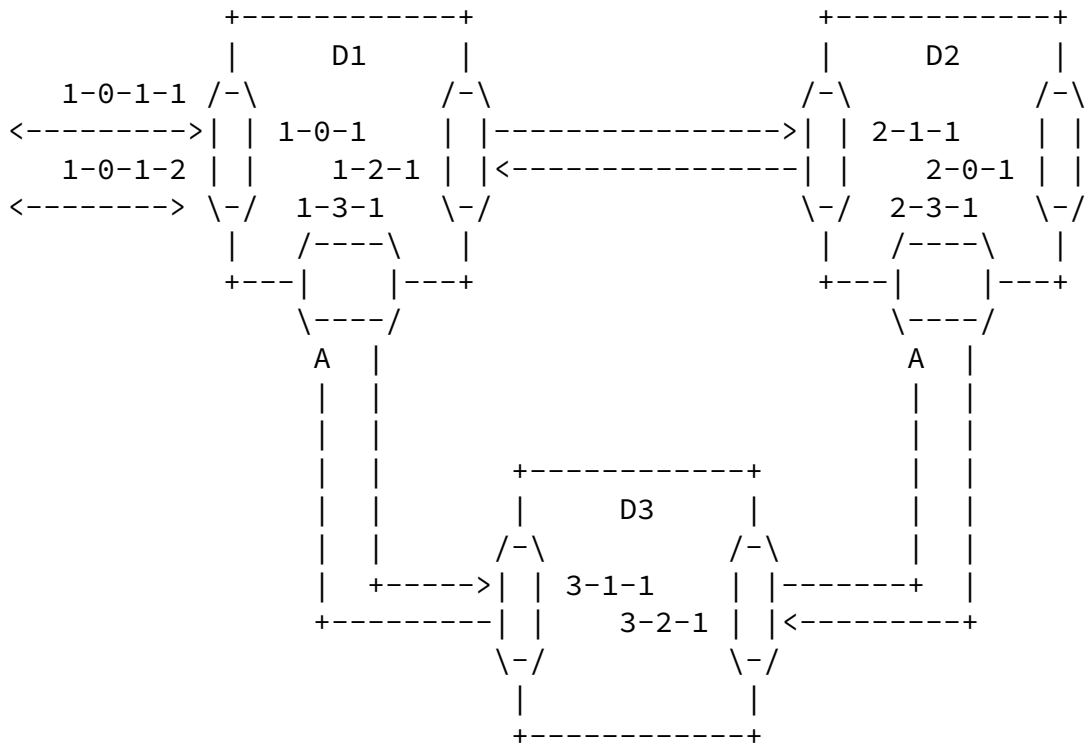   Figure 2: A Network Topology Example

   The corresponding instance data tree is depicted below:

```
{
  "ietf-network:networks": {
    "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",
                  "lag": true,
                  "member-link-tp": [
                    "1-0-1-1",
                    "1-0-1-2"
                  ]
                }
              },
              {
                "tp-id": "1-0-1-1",
                "ietf-l2-topology:l2-termination-point-attributes": {
                  "mac-address": "00:00:5e:00:53:d3"
                }
```

```
      },
      {
        "tp-id": "1-0-1-2",
        "ietf-l2-topology:l2-termination-point-attributes": {
          "mac-address": "00:00:5e:00:53:d4"
        }
      },
      {
        "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::"
      ]
    }
  },
  {
    "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::"
          ]
        }
      },
      {
        "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::"
          ]
        }
      }
    ],
    "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"
                  }
                }
              ]
            }
          ]
        }
      }
```

Acknowledgements

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