A YANG Data Model for Layer-2 Network Topologies
draft-ietf-i2rs-yang-l2-network-topology-08

Abstract

This document defines a YANG data model for Layer 2 network topologies.

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 24, 2020.
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 network topologies by augmenting the generic network and network topology data models with L2 specific topology attributes. A sample example is provided in Appendix B.

This document uses the common YANG types defined in [RFC6991] and adopts the Network Management Datastore Architecture (NMDA [RFC8342]).
The terminology for describing YANG modules is defined in [RFC7950]. The meanings of the symbols used in the tree diagram are defined in [RFC8340].

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.

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 L2 technologies. It can be used to describe both the physical and the logical (virtual) L2 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, descriptions, 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) or other Layer-2 protocols, and some of them may be locally configured.

```
+---------------------+        +---------------------+        +---------------------+
|     ietf-network    |        | ietf-network-topology|        |   ietf-l2-topology   |
| +---------------------+        | +---------------------+        | +---------------------+|
|                         |        |                         |        |                         |
```

Figure 1: L2-topology YANG Module Structure

The structure of "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-network-attributes
    +--rw name?   string
    +--rw flag*   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-vid?       vlan {VLAN}?
    +--rw flag*                 node-flag-type
augment /nw:networks/nw:network/nt:link:
  +--rw l2-link-attributes
    +--rw name?    string
    +--rw flag*    link-flag-type
    +--rw rate?    decimal64
    +--rw delay?   uint32
    +--rw srlg*    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 port-vlan-id?   vlan {VLAN}?
        |   |   +--rw vlan-id-name* [vlan-id] {VLAN}?
        |   |   |   +--rw vlan-id   vlan
        |   |   +--rw vlan-name?   string
        |   +--:(legacy)
        |     +--rw layer-2-address?   yang:phys-address
        |     +--rw encapsulation? identityref
        +--ro tp-state?   enumeration
notifications:
  +--n l2-node-event
    +--ro event-type?
    +--ro node-ref?
    +--ro network-ref?
    +--ro 12-network!
    +--ro 12-node-attributes
  +--n l2-link-event
    +--ro event-type?
    +--ro link-ref?
The L2-topology module augments the generic 'ietf-network' and 'ietf-network-topology' 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 [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 [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.

- The optional L2 technology specific attributes are introduced in this module as Layer-2 features.
4. Layer 2 Topology YANG Module

This module uses the common YANG types defined in [RFC6991] and types defined in [RFC8345].

<CODE BEGINS> file "ietf-l2-topology@2019-06-21.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 "Section 4 of RFC 6991";
  }
  import ietf-yang-types {
    prefix "yang";
    reference "Section 3 of RFC 6991";
  }

  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 <mailto:anders.liu@huawei.com>"
typedef vlan {
    type uint16 {
        range "1..4094";
    }
    description "VLAN ID";
    reference "IEEE 802.1Q";
}

typedef trill-nickname {
    type uint16;
    description "TRILL Nickname";
    reference "RFC 6326";
}

typedef vni {
    type uint32 {
        range "1..16777215";
    }
}
typedef l2-flag-type {
  type identityref {
    base "flag-identity";
  }
  description "Base type for l2 flags";
}
typedef node-flag-type {
  type identityref {
    base "flag-identity";
  }
  description "Node flag attributes";
}
typedef link-flag-type {
  type identityref {
    base "flag-identity";
  }
  description "Link flag attributes";
}
typedef l2-network-event-type {
  type enumeration {
    enum "add" {
      value 0;
      description "An L2 node or link or termination-point has been added";
    }
    enum "remove" {
      value 1;
      description "An L2 node or link or termination-point has been removed";
    }
    enum "update" {
      value 2;
      description "An L2 node or link or termination-point has been updated";
    }
  }
  description "l2 network event type for notifications";
} // l2-topology-event-type

/*
 * Features
 */
feature VLAN {
    description
        "Indicates that the system supports the
        vlan functions.";
}

feature QinQ {
    description
        "Indicates that the system supports the
        qinq functions";
        reference "IEEE 802.1ad.";
}

feature PBB {
    description
        "Indicates that the device supports the
        provider-backbone-bridging functions";
        reference "IEEE 802.1ah.";
}

feature VPLS {
    description
        "Indicates that the device supports the
        VPLS functions.";
        reference "RFC 4761, RFC 4762";
}

feature TRILL {
    description
        "Indicates that the device supports the
        TRILL functions.";
        reference "RFC 6325";
}

feature VXLAN {
    description
        "Indicates that the device supports the
        VXLAN functions.";
        reference "RFC 7348";
}

/*
 * Identities
 */

identity flag-identity {
    description "Base type for flags.";
}
identity encapsulation-type {
    description
        "Base identity from which specific encapsulation
types are derived."
}

identity eth-encapsulation-type {
    base encapsulation-type;
    description
        "Base identity from which specific Ethernet
encapsulation types are derived."
}

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
        "PBB encapsulation."
}

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

identity vpls {
    base eth-encapsulation-type;
    description
        "VPLS encapsulation."
}
identity vxlan {
    base eth-encapsulation-type;
    description
        "VXLAN encapsulation.";
}

identity frame-relay {
    base encapsulation-type;
    description
        "Frame Relay encapsulation.";
}

identity ppp {
    base encapsulation-type;
    description
        "PPP encapsulation";
}

identity hdlc {
    base encapsulation-type;
    description
        "HDLC encapsulation.";
}

identity atm {
    base encapsulation-type;
    description
        "Base identity from which specific ATM
            encapsulation types are derived.";
}

identity pwe3 {
    base encapsulation-type;
    description
        "Base identity from which specific PW
            encapsulation types are derived.";
}

/*
 * Groupings
 */

grouping l2-network-type {


description "Identify 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-network-attributes {
    description "L2 Topology scope attributes";
    container l2-network-attributes {
        description "Containing L2 network attributes";
        leaf name {
            type string;
            description "Name of the L2 network."
        }
        leaf-list flag {
            type l2-flag-type;
            description "L2 network flags"
        }
    }
}


grouping l2-node-attributes {
    description "L2 node attributes";
    container l2-node-attributes {
        description "Containing 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 "System management address."
        }
        leaf sys-mac-address {
            type yang:mac-address;
            description "System MAC-address."
        }
        leaf management-vid {
            if-feature VLAN;
            type vlan;
        }
    }
}
description "System management VID.";
}
leaf-list flag {
  type node-flag-type;
  description "Node operational flags.";
}
}
} // grouping l2-node-attributes

grouping l2-link-attributes {
  description "L2 link attributes";
  container l2-link-attributes {
    description "Containing L2 link attributes.";
    leaf name {
      type string;
      description "Link name.";
    }
    leaf-list flag {
      type link-flag-type;
      description "Link flags.";
    }
    leaf rate {
      type decimal64 {
        fraction-digits 2;
      }
      description "Link rate.";
    }
    leaf delay {
      type uint32;
      description "Link delay in microseconds.";
    }
    leaf-list srlg {
      type uint32;
      description "List of Shared Risk Link Groups this link belongs to.";
      reference "RFC 4202";
    }
  }
} // grouping l2-link-attributes

grouping l2-termination-point-attributes {
  description "L2 termination point attributes";
  container l2-termination-point-attributes {
    description "Containing L2 TP attributes";
    leaf description {
      type string;
description "Port description."
}

leaf maximum-frame-size {
  type uint32;
  description "Maximum frame size."
}

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 port-vlan-id {
    if-feature VLAN;
    type vlan;
    description "Port VLAN ID is the VLAN id that will be assigned to any untagged frames entering the switch on the specific port."
  }
}

list vlan-id-name {
  if-feature VLAN;
  key "vlan-id";
  description "Interface configured VLANs."
  leaf vlan-id {
    type vlan;
    description "VLAN ID."
  }
  leaf vlan-name {
    type string;
    description "VLAN name."
  }
}
case legacy {
    leaf layer-2-address {
        type yang:phys-address;
        description "Interface Layer 2 address.";
    }
    leaf encapsulation {
        type identityref {
            base encapsulation-type;
        }
        description "Encapsulation type of this termination point.";
    }
} //case legacy such as atm, ppp, hdlc, etc.

} //choice termination-point-type

leaf tp-state {
    type enumeration {
        enum in-use {
            value 0;
            description "the termination point is in forwarding state.";
        }
        enum blocking {
            value 1;
            description "the termination point is in blocking state.";
        }
        enum down {
            value 2;
            description "the termination point is in down state.";
        }
        enum others {
            value 3;
            description "the termination point is in other state.";
        }
    }
    config false;
    description "State of the termination point";
} // grouping l2-termination-point-attributes
/*  * Data nodes  */

augment "nw:networks/nw:network/nw:network-types" {
    description
    "Introduce 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-network-attributes;
}

desc
"Augmentation parameters apply only for networks with L2 topology."
}

description
"Augment L2 topology termination point information.");
uses 12-termination-point-attributes;
}

/* * Notifications */

notification l2-node-event {

description "Notification event for L2 node";
leaf event-type {

type 12-network-event-type;

description "Event type.";
}
uses nw:node-ref;
uses 12-network-type;
uses 12-node-attributes;
}

notification l2-link-event {

description "Notification event for L2 link.";
leaf event-type {

type 12-network-event-type;

description "Event type";
}
uses nt:link-ref;
uses 12-network-type;
uses 12-link-attributes;
}

notification l2-termination-point-event {

description "Notification event for L2 termination point.";
leaf event-type {

type 12-network-event-type;

description "Event type";
}
uses nt:tp-ref;
uses 12-network-type;
uses 12-termination-point-attributes;
}

} // module l2-topology
<CODE ENDS>
5. IANA Considerations

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

Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

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 [RFC7950] within the "YANG Parameters" registry.

name: ietf-l2-topology
prefix: l2t
reference: RFC XXXX

name: 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.

In general, Layer 2 network topologies are system-controlled and provide ephemeral topology information. In an NMDA-compliant server, they are only part of <operational> which provides read-only access
to clients, they are less vulnerable. That said, the YANG module does in principle allow information to be configurable.

The Layer 2 topology module define 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, leading to disruption of services provided via this topology might be disrupted. For those reasons, it is important that the NETCONF access control model 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 in the ietf-network module:

- **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, such as the maximum-frame-size.
7. Acknowledgements

The authors would like to acknowledge the comments and suggestions received from Susan Hares, Alia Atlas, Juergen Schoenwaelder, Mach Chen, Alexander Clemm and Sriганesh Kini.

8. References

8.1. Normative References


8.2. Informative References


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 represent the operational state of layer 2 network topologies. The module ietf-l2-topology-state mirrors the module ietf-l2-topology defined earlier in this document. 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 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@2019-06-21.yang"
module ietf-l2-topology-state {
    yang-version 1.1;
    prefix "l2t-s";

    import ietf-network-state {
        prefix "nw-s";
    }

    import ietf-network-topology-state {
        prefix "nt-s";
    }

    import ietf-l2-topology {
        prefix "l2t";
    }

    organization

/*
 * Data nodes
 */

augment "/nw-s:networks/nw-s:network/nw-s:network-types" {
  description
    "Introduce 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: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 L2 at the node
      level";
   uses l2t:l2-node-attributes;
}

augment "/nw-s:networks/nw-s:network/nt-s:link" {
   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 "Augment 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:networks/nw-s:network/nw-s:network-types/"+
   "l2t-s:l2-network" {
      description
         "Augmentation parameters apply only for networks
         with L2 topology";
   }
   description "Augment 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 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;
}

} // module l2-topology-state

<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.

![Diagram of network topology example]

Figure 2. A Network Topology Example

The corresponding instance data tree is depicted as below. Note that some lines have been wrapped to adhere to the 72-character line limitation of RFCs.

```json
{
  "ietf-network:networks": {
    "network": [
      {
        "network-types": {
          "ietf-l2-topology:l2-network": {}
        },
        "network-id": "l2-topo-example",
        "node": [
          {
            "node-id": "D1",
            "termination-point": [
              {
                "tp-id": "1-0-1",
                "ietf-l2-topology:12-termination-point-attributes": {
                }
              }
            ]
          }
        ]
      }
    ]
  }
}
```


```json
{
  "node-id": "D1",
  "termination-point": [
    {
      "tp-id": "1-2-1",
      "ietf-l2-topology:
        l2-termination-point-attributes": {
        }
    },
    {
      "tp-id": "1-3-1",
      "ietf-l2-topology:
        l2-termination-point-attributes": {
        }
    }
  ],
  "ietf-l2-topology:l2-node-attributes": {
    "management-address": ["10.1.1.1"]
  }
},
{
  "node-id": "D2",
  "termination-point": [
    {
      "tp-id": "2-0-1",
      "ietf-l2-topology:
        l2-termination-point-attributes": {
          "mac-address": "A2:B2:C3:D4:E5:F0"
        }
    },
    {
      "tp-id": "2-1-1",
      "ietf-l2-topology:
        l2-termination-point-attributes": {
          "mac-address": "A2:B2:C3:D4:E5:F2"
        }
    },
    {
      "tp-id": "2-3-1",
      "ietf-l2-topology:
        l2-termination-point-attributes": {
        }
    }
  ],
  "ietf-l2-topology:l2-node-attributes": {
    "management-address": ["10.1.1.2"]
  }
}
```


),

{ "node-id": "D3",
 "termination-point": [
  { "tp-id": "3-1-1",
    "ietf-l2-topology:12-termination-point-attributes": {
      "mac-address": "A3:B2:C3:D4:E5:F0"
    }
  },
  { "tp-id": "3-2-1",
    "ietf-l2-topology:12-termination-point-attributes": {
      "mac-address": "A3:B2:C3:D4:E5:F1"
    }
  }
],
 "ietf-l2-topology:12-node-attributes": {
  "management-address": ["10.1.1.3"]
}
}

},

"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:12-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:12-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:12-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:12-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