

I2RS Working Group
Internet-Draft
Intended status: Standards Track

Xian Zhang
Baoquan Rao
Huawei
Xufeng Liu
Ericsson

Expires: September 9, 2015

March 9, 2015

A YANG Data Model for Layer 1 Network Topology

draft-zhang-i2rs-l1-topo-yang-model-01.txt

Abstract

This draft describes a YANG data model to manipulate the topologies of a layer 1 network. It is independent of data plan technologies and control plane protocols. It can be augmented to include technology-specific data, such as for Optical Transport Networks (OTN).

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on September 9, 2015.

Copyright Notice

Copyright (c) 2015 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 (<http://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	2
2. Conventions used in this document.....	3
3. Terminology and Notations.....	3
4. YANG Data Model for Layer 1 Topology.....	4
4.1. YANG Tree	4
4.1.1. The node and link list.....	5
4.1.2. Notification.....	5
4.2. YANG Code	5
5. Security Considerations	21
6. Manageability Considerations	21
7. IANA Considerations	21
8. Acknowledgements	21
9. References	22
9.1. Normative References	22
9.2. Informative References	22
10. Contributors' Addresses	22
11. Authors' Addresses	22

1. Introduction

This document defines a data model of a layer one network topology, using YANG [RFC6020]. The model can be used by an application via the I2RS interface [draft-ietf-i2rs-architecture], in the following ways (but not limited to):

- o to obtain a whole view of the network topology information of its interest;
- o to receive notifications with regard to the information of the change of the network topology of its interest;

- o to enforce the establishment/update of a network topology with the characteristic specified in the data model;

This model is confined to describe layer 1 networks, but it is data plane technology independent and can be augmented to specify the topology for networks such as Optical Transport networks (OTN), Synchronous Digital Network/ (SDH/SONET) DWDM (Dense Wavelength Division Multiplexing).

[Editor's Note: The authors are aware that there are other drafts closely relating to this draft. Coordination works have been undergoing to get these drafts aligned. The authors are working on obtaining layer one topology by augmenting the data model proposed in draft-clemm-i2rs-yang-network-topo in the next version of this draft.]

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

3. Terminology and Notations

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in the YANG data tree presented later in this draft is defined in [ietf-netmod-rfc6087bis]. They are provided below for reference.

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

4. YANG Data Model for Layer 1 Topology

4.1. YANG Tree

```

module: ietf-layer1topology
  +--rw layer-one-topology
    +--rw topology* [topology-id]
      +--rw topology-id          topology-id
      +--rw name?                 string
      +--rw supporting-topology* [topo-ref]
        | +--rw topo-ref         leafref
      +--rw node* [node-id]
        | +--rw node-id          node-id
        | +--rw interface* [interface-id]
        |   | +--rw interface-id  interface-id
        |   | +--rw interface-name? if:interface-state-ref
        |   | +--rw adaptation-capability
        | +--rw connectivity-matrix* [id]
        |   | +--rw id            uint32
        |   | +--rw type?         enumeration
        |   | +--rw in-interface* [interface-ref]
        |   |   | +--rw interface-ref leafref
        |   | +--rw out-interface* [interface-ref]
        |   |   | +--rw interface-ref leafref
        |   | +--rw dir?         enumeration
      +--rw link* [link-id]
        +--rw link-id            link-id
        +--rw local
          | +--rw local-node      leafref
          | +--rw local-interface leafref
        +--rw remote
          | +--rw remote-node     leafref
          | +--rw remote-interface leafref
        +--rw supporting-path* [supporting-path-index]
          | +--rw supporting-path-index uint32
          | +--rw topo-ref?          leafref
          | +--rw server-path-identifier
          | +--rw server-path-srlg
          |   | +--rw srlg-values* [srlg-value]
          |   | +--rw srlg-value   uint32
        +--rw attributes
          +--ro information-source? enumeration
          +--ro credibility-preference? uint16
          +--rw admin-status?         enumeration
          +--ro oper-status?          enumeration
          +--rw area-id?              binary
          +--rw max-link-bandwidth?   decimal64

```

```

        +---rw unreserved-bandwidth* [priority]
        |   +---rw priority      uint8
        |   +---rw bandwidth?   decimal64
        +---ro distance?                uint32
        +---rw te-metric?                uint32
        +---rw link-protection-type?     enumeration
        +---rw switching-capability?     switching-
capabilities
        +---rw encoding?                encoding-types
        +---rw switching-capability-specific
        +---rw srlg
            +---rw srlg-values* [srlg-value]
            +---rw srlg-value      uint32
notifications:
+---n link-failure
|   +---ro topology-id      leafref
|   +---ro link-id         leafref
|   +---ro admin-status?   leafref
|   +---ro oper-status     leafref
+---n node-failure
    +---ro topology-id      leafref
    +---ro link-id         leafref

```

4.1.1. The node and link list

The Layer One Topology module contains all the nodes and links information pertaining to a layer one network. The node is identified by the node-id, which is unique within the network. Within the nodes, all the interfaces pertaining to this node and their potential capabilities/constraints SHOULD be present. Besides this, the constraints associated with a node as a whole SHOULD also be present, such as the connectivity constraints introduced due to abstraction or due to the hardware limitations. The link is identified by the link-id, which is unique within a node. It includes the association with nodes as well as interfaces. Moreover, it includes information that is of interest to the I2RS client, for purposes, such as path computation, monitoring etc.

4.1.2. Notification

Two types of notifications are introduced: node failure and link failure.

4.2. YANG Code

<CODE BEGINS> file "lltopo.yang"

```
module ietf-layer1topology {
  yang-version 1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-layer1topology";
  prefix "lltopo";

  import ietf-inet-types {
    prefix "inet";
  }
  import ietf-interfaces {
    prefix "if";
  }

  organization
    "Internet Engineering Task Force (IETF) I2RS WG";
  contact
    "ID-draft editor: zhang.xian@huawei.com";

  description
    "This module defines a data-plan technology/protocol
    independent Layer One topology data model.";

  revision 2015-03-09 {
    description
      "Initial version.";
    reference
      "draft-zhang-i2rs-ll-topo-yang-model-01.txt";
  }

  /*
  * Typedefs
  */

  typedef topology-id {
    type inet:uri;
    description "the identifier for a topology";
  }

  typedef node-id {
    type inet:ip-address;
    description
      "the identifier for a node";
  }

  typedef interface-id {
    type union {
```

```
    type inet:ip-address; // IPv4 or IPv6 address
    type int32;           // Un-numbered
}
description
  "the identifier of an interface within a node, supporting both
  numbered/unnumbered";
}

typedef link-id {
  type inet:ip-address; // IPv4 or IPv6 address
  description "the identifier of a link";
}

typedef switching-capabilities {
  type enumeration {
    enum "psc-1" {
      value 1;
      description
        "Packet-Switch Capable-1 (PSC-1)";
    }
    enum "evpl" {
      value 30;
      description
        "Ethernet Virtual Private Line (EVPL)";
    }
    enum "pbb-te" {
      value 40;
      description
        "802_1 PBB-TE";
    }
    enum "l2sc" {
      value 51;
      description
        "Layer-2 Switch Capable (L2SC)";
    }
    enum "tdm" {
      value 100;
      description
        "Time-Division-Multiplex Capable (TDM)";
    }
    enum "otn-tdm" {
      value 110;
      description
        "OTN-TDM Capable";
    }
  }
}
```

```
    enum "lsc" {
        value 150;
        description
            "Lambda-Switch Capable (LSC)";
    }
    enum "fsc" {
        value 200;
        description
            "Fiber-Switch Capable (FSC)";
    }
}

description
    "Switching capability of an interface.
    Only a subset of the above-mentioned values are applicable
    to Layer 1 network.
    Here it is included for completeness and will later be
    updated if a base model is augmented to create layer 1
    network topology YANG data model.";

reference
    "The definition of switching types, their values and the
    relevant RFCs can be found at:
    http://www.iana.org/assignments/gmpls-sig-parameters/gmpls-sig-parameters.xhtml#gmpls-sig-parameters-3";
}

typedef encoding-types {
    type enumeration {
        enum "packet" {
            value 1;
            description "Packet";
        }
        enum "ethernet" {
            value 2;
            description "Ethernet";
        }
        enum "pdh" {
            value 3;
            description "PDH";
        }
        enum "sdh-sonet" {
            value 5;
            description "SDH/SONET";
        }
        enum "digital-wrapper" {
            value 7;
        }
    }
}
```



```

        description "Digital Wrapper";
    }
    enum "lambda" {
        value 8;
        description "Lambda(photonic)";
    }
    enum "fiber" {
        value 9;
        description "Fiber";
    }
    enum "fiber-channel" {
        value 11;
        description "FiberChannel";
    }
    enum "oduk" {
        value 12;
        description
            "G.709 OKUk (Digital Path)";
    }
    enum "optical-channel" {
        value 13;
        description "G.709 Optical Channel";
    }
    enum "line" {
        value 14;
        description "Line (e.g., 8B/10B)";
    }
}
description
    "The encoding type supported by an interface or link.
    Not all encoding types are applicable to Layer one network
    nodes. They are included here for completeness and will be
    updated if a base model is available to augment
    so as to build a layer-one specific YANG data model.";
reference
    "The definition of encoding types, their values and the
    relevant RFCs can be found at http://www.iana.org/
    assignments/gmpls-sig-parameters/gmpls-sig-parameters.xhtml#
    gmpls-sig-parameters-3";
}

/*
 * Groupings
 */

grouping srlg-attribute {
    description

```

```
    "Shared Risk Link Group Attributes";
  reference
    "RFC 4203: OSPF Extensions in Support of Generalized
    Multi-Protocol Label Switching (GMPLS)";
  list srlg-values {
    key "srlg-value";
    leaf srlg-value {
      type uint32;
      description "SRLG value";
    }
  }
  description
    "the SRLG value list";
}

/*
 * Configuration data nodes
 */

container layer-one-topology {
  description
    "this container holds all the inforamtion to layer
    one network. It includes one or multiple topologies";

  list topology {
    key "topology-id";

    description
      "This contains all the information to one topoogy";

    leaf topology-id {
      type topology-id;
      description "topology identifier";
    }

    leaf name {
      type string;
      description "topology name";
    }

    list supporting-topology {
      key "topo-ref";
      leaf topo-ref {
        type leafref {
          path "/layer-one-topology/topology/topology-id";
        }
        description

```

```
        "a Layer-One network might be supported by a lower
        layer network and this is a pointer to the supporting
        topology if there is one";
    }
    description "underlaying topology information";
}

list node {
    key "node-id";
    description "the list of nodes within the topology";

    leaf node-id {
        type node-id;
        description "node identifier";
    }

    list interface {
        key "interface-id";
        leaf interface-id {
            type interface-id;
            description "interface identifier";
        }
        leaf interface-name {
            type if:interface-state-ref;
            description
                "Name of the incoming interface.";
        }
        container adaptation-capability {
            description
                "TBD -to add for technology specific information";
        }
        description "interface list pertaining to a node";
    }
}

list connectivity-matrix {
    key "id";

    description
        "This describes the connectivity constraints within
        a node in the network. It can be one matrix or a set
        of matrixes. Further details, read the reference
        provided below.";
    reference
        "https://tools.ietf.org/html/draft-ietf-ccamp-general-
        -constraint-encode-16 Section 2.1";

    leaf id {
```

```
        type uint32;
        description "matrix id";
    }
    leaf type {
        type enumeration {
            enum fixed {
                value 0;
                description "Fixed";
            }
            enum dynamic {
                value 1;
                description "Dynamic/changeable";
            }
        }
    }
    description
        "This field describes the attribute of a
        connectivity matrix, i.e., whether it is
        fixed or switched.";
}
list in-interface {
    key "interface-ref";

    description
        "This list describes a (sub)-set of ingoing
        interfaces within a node that may have
        connectivity constraints.
        Note: directionality may not be relevant
        and it is decided by the dir parameter.";

    leaf interface-ref {
        type leafref {
            path "/layer-one-topology/topology/node/" +
                "interface/interface-id";
        }
        description "reference to an incoming interface";
    }
}
list out-interface {
    key "interface-ref";

    description
        "This list describes a (sub)-set of outgoing
        interfaces within a node that may have
        connectivity constraints.
        Note: directionality may not be relevant and
        it is decided by the dir parameter.";
```

```

        leaf interface-ref {
            type leafref {
                path "/layer-one-topology/topology/node/"+
                    "interface/interface-id";
            }
            description "reference to an outgoing interface";
        }
    }
    leaf dir{
        type enumeration{
            enum "uni-dir"{
                description
                    "the matrix is unidirectional.";
            }
            enum "bi-dir"{
                description
                    "this matrix is bidirecdtional.";
            }
        }
        description
            "the directionality attribute of a connc. matrix.";
    }
}
} // end of node data node

list link {
    key "link-id";

    description "list of the links within a topology";

    leaf link-id {
        type link-id;
        description
            "remaining issue: if there is no IP addresses
            associated with this link, what would be the key?";
    }

    container local {
        description "near end information for this link";

        leaf local-node {
            type leafref {
                path "/lltopo:layer-one-topology/topology"+
                    "/node/node-id";
            }
            mandatory true;
            description "refence to the local node";
        }
    }
}

```

```

    }
    leaf local-interface {
        type leafref {
            path "/lltopo:layer-one-topology/topology/node/"
                + "interface/interface-id";
        }
        mandatory true;
        description "reference to the local interface";
    }
}
container remote {
    description "far end information of this link";

    leaf remote-node {
        type leafref {
            path "/lltopo:layer-one-topology/topology"+
                "/node/node-id";
        }
        mandatory true;
        description "reference to the remote node";
    }
    leaf remote-interface {
        type leafref {
            path "/lltopo:layer-one-topology/topology/node/"
                + "interface/interface-id";
        }
        mandatory true;
        description "reference to the remote interface";
    }
}
list supporting-path {
    key "supporting-path-index";

    description
        "information pertaining to the underlying path if
        there is any";

    leaf supporting-path-index {
        type uint32;
        description "the identifier of the supporting path";
    }
    leaf topo-ref {
        type leafref {
            path "/lltopo:layer-one-topology/" +
                "topology/topology-id";
        }
        description "reference to the underlying topology";
    }
}

```

```
    }
    container server-path-identifier {
        description "TBD";
    }
    container server-path-srlg {
        uses srlg-attribute;
        description "the SRLG values of the server path";
    }
}

container attributes {

    description "additional information of the link";

    leaf information-source {
        type enumeration {
            enum "unknown" {
                description "The source is unknown";
            }
            enum "locally-configured" {
                description "Configured TE link";
            }
            enum "ospfv2" {
                description "OSPFv2";
            }
            enum "ospfv3" {
                description "OSPFv3";
            }
            enum "isis" {
                description "ISIS";
            }
        }
    }
    config false;

    description
        "Indicates the source of the information about
        the link. remaining issue: if configuration of
        a link is allowed, what additional types are
        needed to add?";
}
leaf credibility-preference {
    type uint16;
    config false;
    description "the level of credibility";
}
leaf admin-status {
    type enumeration {
```

```
        enum up {
            value 1;
            description "up";
        }
        enum down {
            value 2;
            description "down";
        }
        enum testing {
            value 3;
            description "testing - in some test mode.";
        }
    }
    description
        "The administrative state of the link.";
    reference
        "RFC2863: The Interfaces Group MIB.";
}
leaf oper-status {
    type enumeration {
        enum up {
            value 1;
            description "up";
        }
        enum down {
            value 2;
            description "down";
        }
        enum testing {
            value 3;
            description "testing - in some test mode.";
        }
        enum unknown {
            value 4;
            description "unknown - status cannot be
                determined for some reason.";
        }
        enum dormant {
            value 5;
            description "dormant";
        }
    }
}
config false;
description
    "The current operational state of the link.";
reference
    "RFC2863: The Interfaces Group MIB.";
```



```
}
leaf area-id {
  type binary {
    length 1..13;
  }
  description
    "This object indicates the area identifier of
    the IGP. If OSPF is used to advertise LSA,
    this represents an ospfArea. If IS-IS is used,
    this represents an area address.
    Otherwise, this is zero.";
  reference
    "RFC4920: Crankback Signaling Extensions for MPLS
    and GMPLS RSVP-TE.";
}

leaf max-link-bandwidth {
  type decimal64 {
    fraction-digits 2;
  }
  description
    "the max bandwidth supported by this link";
}

list unreserved-bandwidth {
  key "priority";
  max-elements "8";

  description
    "This describes the unreserved bandwidth (in
    Bytes/second) on a level basis ( level 0-7).";

  leaf priority {
    type uint8{
      range "0..7";
    }
    description "priority level";
  }
  leaf bandwidth {
    type decimal64 {
      fraction-digits 2;
    }
    description "badnwidth per priority";
  }
}

leaf distance {
```

```
    type uint32;
    units "kilometers";

    config false;
    description
        "the distance this link spans.";
}
leaf te-metric {
    type uint32;
    description "the metric supported by the link";
}

leaf link-protection-type {
    type enumeration {
        enum "extra-traffic" {
            value 1;
            description "Extra traffic";
        }
        enum "unprotected" {
            value 2;
            description "unprotected";
        }
        enum "shared" {
            value 4;
            description "Shared";
        }
        enum "1-for-1" {
            value 8;
            description "Dedicated one for one protection";
        }
        enum "1-plus-1" {
            value 16;
            description "Dedicated one plus one protection";
        }
        enum "enhanced" {
            value 32;
            description "a protection type that is
                more reliable than Dedicated 1+1,
                e.g., 4 fiber BLSR/MS-SPRING.";
        }
    }
}
description
    "Link Protection Type configured for this link";
reference
    "RFC3471: Generalized Multi-Protocol Label
        Switching (GMPLS) Signaling Functional
        Description.";
```

```

    }

    leaf switching-capability {
        type switching-capabilities;
        description
            "the switching capability supported by the link";
    }
    leaf encoding {
        type encoding-types;
        description
            "the encoding type supported by this link.";
    }

    container switching-capability-specific {
        description
            "TBD - to add for technology specific information";
    }
    container srlg {
        uses srlg-attribute;
        description " the SRLG values of a link";
    }
    } // end of link attributes
} // end of link leaf data node
}
} // end of configuring data nodes

/*
* notifications - only provide operational change information.
* reply to topology/node/link creation is acked via rpc-reply.
*/

notification link-failure {
    leaf topology-id {
        type leafref {
            path "/layer-one-topology/topology/topology-id";
        }
        mandatory true;
        description "";
    }
    leaf link-id {
        type leafref {
            path
                "/layer-one-topology/topology[topology-id="+
                "current ()/../../topology-id]/link/link-id";
        }
        mandatory true;
        description "";
    }
}

```

```

    }
    leaf admin-status {
      type leafref {
        path
          "/layer-one-topology/topology/link[link-id =" +
          "current()/../link-id]/attributes/admin-status";
      }
      description "";
    }
    leaf oper-status {
      type leafref {
        path
          "/layer-one-topology/topology/" +
          "link[link-id = current()/../link-id]"
          + "/attributes/oper-status";
      }
      mandatory true;
      description "";
    }
  }
  description
    "link failure information";
} //notification

notification node-failure {
  leaf topology-id {
    type leafref {
      path "/layer-one-topology/topology/topology-id";
    }
    mandatory true;
    description "";
  }
  leaf link-id {
    type leafref {
      path
        "/layer-one-topology/topology[topology-id= "
        + "current ()/../topology-id]/node/node-id";
    }
    mandatory true;
    description "";
  }
  description
    "node failure information";
} //notification
} //module

<CODE ENDS>

```

5. Security Considerations

Since the data model defined in this draft is manipulated via the I2RS interface. The security concerns mentioned in [draft-ietf-i2rs-architecture] also applies to this draft.

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which 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.

[Editor's note: to List specific subtrees and data nodes and their sensitivity/vulnerability.]

6. Manageability Considerations

TBD.

7. IANA Considerations

TBD.

8. Acknowledgements

The initial YANG model specified in this draft is based on draft-clemm-i2rs-yang-network-topo but it is modified according to the features of the layer one networks.

We would like to thank the authors of the above mentioned draft for their helpful discussion during the creation of this draft.

9. References

9.1. Normative References

- [RFC2119] S. Bradner, "Key words for use in RFCs to indicate requirements levels", RFC 2119, March 1997.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

9.2. Informative References

- [draft-ietf-i2rs-architecture] Atlas, A., Halpern, J., Hares, S., Ward, D., Nadeau T., "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-08, work in progress, January 2015;
- [draft-clemm-i2rs-yang-network-topo] Clemm A., Medved J., Tkacik T., Varga R., et al, "A YANG Data Model for Network Topologies", draft-clemm-i2rs-yang-network-topo-01, work in progress, October 2014;
- [ietf-netmod-rfc6087bis] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", draft-ietf-netmod-rfc6087bis-01, work in progress, October 2014.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC6241, June 2011.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

10. Contributors' Addresses

TBD.

11. Authors' Addresses

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Baoquan Rao
Huawei Technologies
raobaoquan@huawei.com

Xufeng Liu
Ericsson
xufeng.liu@ericsson.com

