

I2RS Working Group
Internet-Draft
Intended status: Standards Track

Xian Zhang
Baoquan Rao
Huawei

Expires: August 13, 2015

February 12, 2015

A YANG Data Model for Layer 1 Network Topology

[draft-zhang-i2rs-l1-topo-yang-model-00.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/1id-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 August 13, 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	3
4.1.	YANG Tree	3
4.1.1.	Notification.....	5
4.2.	YANG Code	5
5.	Security Considerations.....	18
6.	Manageability Considerations.....	18
7.	IANA Considerations	18
8.	Acknowledgements	18
9.	References	18
9.1.	Normative References.....	18
9.2.	Informative References.....	19
10.	Contributors' Addresses.....	19
11.	Authors' Addresses	19

[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).

[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.]

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: layer-one-topology
  +--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 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 te-attributes
    +--rw link-index?                uint64
    +--rw information-source?         enumeration
    +--rw credibility-preference?     uint16
    +--rw admin-status?               enumeration
    +--rw oper-status?                enumeration
    +--rw area-id?                    binary
    +--rw max-link-bandwidth?         decimal64
    +--rw unreserved-bandwidth* [priority]
    | +--rw priority                uint8
    | +--rw bandwidth?              decimal64
    +--rw distance?                  uint32

```

```

    +--rw te-default-metric?                uint32
    +--rw link-protection-type?              enumeration
    +--rw interface-switching-capability* [switching-
capability]
        | +--rw switching-capability switching-capabilities
        | +--rw encoding? encoding-types
        | +--rw max-lsp-bandwidth* [priority]
        | | +--rw priority                uint8
        | | +--rw bandwidth?              decimal64
        | +--rw switching-capability-specific-information!
    +--rw srlg
        +--rw srlg-values* [srlg-value]
        +--rw srlg-value      uint32

```

[4.1.1. Notification](#)

[TBD]

[4.2. YANG Code](#)

<CODE BEGINS>

```

module layer-one-topology {
  yang-version 1;

  namespace "urn:ietf:params:xml:ns:yang:layer-one-topology";
  prefix l1topo;

  import ietf-inet-types {
    prefix inet;
  }

  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 2015-02-11 {
    description
      "Initial version";

    reference
      "TBD: draft-zhang-i2rs-l1-topo-yang-model-00.txt";
  }
}

```

```
/*
 * Typedefs
 */

typedef topology-id {
    type inet:uri;
}

typedef node-id {
    type inet:ip-address;
}

typedef interface-id {
    type union {
        type inet:ip-address; // IPv4 or IPv6 address
        type int32;           // Unnumbered
    }
    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 (i.e., RFC3471, 4328) 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 ODUk (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 (RFC3471, RFC4328, RFC6004) can be found at
http://www.iana.org/assignments/gmpls-sig-parameters/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;
        }
    }
}

/*
 * Configuration data nodes
 */

container layer-one-topology {
    list topology {
        key "topology-id";
        leaf topology-id {
            type topology-id;
        }
    }
}
```



```
    }

    leaf name {
      type string;
    }

    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 suporting topology if there is
one";
      }
    }

    list node {
      key "node-id";
      leaf node-id {
        type node-id;
      }
      list interface {
        key "interface-id";
        leaf interface-id {
          type interface-id;
        }
        container adaptation-capability {
          presence "to define adaptation capability for different
technologies.";
          description
            "TBD";
        }
      }
      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;
}
leaf type {
  type enumeration {
    enum fixed {
      value 0;
    }
    enum dynamic {
      value 1;
    }
  }

  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";
      }
    }
  }
list out-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";
      }
    }
  }
```



```
    }
  }
  leaf dir{
    type enumeration{
      enum "uni-dir"{
        description
          "the matrix is unidirectional.";
      }
      enum "bi-dir"{
        description
          "this matrix is bidirecdtional.";
      }
    }
  }
}

list link {
  key "link-id";
  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 {
  leaf local-node {
    type leafref {
      path "/l1topo:layer-one-topology/topology/node/node-id";
    }
  }
  leaf local-interface {
    type leafref {
      path "/l1topo:layer-one-
topology/topology/node/interface/interface-id";
    }
  }
}

container remote {
  leaf remote-node {
    type leafref {
      path "/l1topo:layer-one-topology/topology/node/node-
id";
    }
  }
}
```



```
        leaf remote-interface {
            type leafref {
                path "/l1topo:layer-one-
topology/topology/node/interface/interface-id";
            }
        }
    }
    list supporting-path {
        key "supporting-path-index";
        leaf supporting-path-index {
            type uint32;
        }
        leaf topo-ref {
            type leafref {
                path "/l1topo:layer-one-topology/topology/topology-id";
            }
        }
        container server-path-identifier {
            presence "Define server path identifier.";
        }
        container server-path-srlg {
            uses srlg-attribute;
        }
    }
}

container te-attributes {
    leaf link-index {
        type uint64;
    }
    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;
    }
    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;
    }
    }
    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,
it can identify either a OSPF or ISIS area.";
        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";
      }
    }
    leaf bandwidth {
      type decimal64 {
        fraction-digits 2;
      }
    }
  }

  leaf distance {
    type uint32;
    units "kilometers";

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

  leaf link-protection-type {
    type enumeration {
      enum "extra-traffic" {
        description "Extra traffic";
      }
      enum "unprotected" {
        description "unprotected";
      }
      enum "shared" {
        description "Shared";
      }
      enum "1-for-1" {
        description "Dedicated one for one protection";
      }
      enum "1-plus-1" {
        description "Dedicated one plus one protection";
      }
    }
  }
```



```
        enum "enhanced" {
            description "a protection type that is more reliable
than Dedicated 1-plus-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.";
    }
    list interface-Switching-capability {
        key "switching-capability";
        leaf switching-capability {
            type switching-capabilities;
        }
        leaf encoding {
            type encoding-types;
            description
                "the encoding type supported by this link.";
        }
    }
    list max-lsp-bandwidth {
        key "priority";
        max-elements "8";
        leaf priority {
            type uint8 {
                range "0..7";
            }
        }
        leaf bandwidth {
            type decimal64 {
                fraction-digits 2;
            }
        }
        description
            "remaining issue: to check how is used.";
    }
    container switching-capability-specific-information {
        presence "to define switch capability specific
information for technology specific network (e.g., OTN).";
    }
    }
    container srlg {
        uses srlg-attribute;
    }
    }
}
```



```
}  
}  
}  
<CODE ENDS>
```

5. Security Considerations

The YANG data model defined in this draft itself does not introduce additional security concerns besides those mentioned in [[draft-ietf-i2rs-architecture](#)].

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.

[Editor's note: Further modification of this data model will be seeking ways to get aligned by augmenting.]

We would like to thank the authors of the above mentioned draft, especially Xufeng Liu, for his 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.

[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.

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;

10. Contributors' Addresses

11. Authors' Addresses

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

Baoquan Rao
Huawei Technologies
raobaoquan@huawei.com