Network Working Group
Internet Draft
Intended status: Standards Track

Xufeng Liu
Ericsson
Vishnu Pavan Beeram
Juniper Networks
Alexander Clemm
Cisco
Igor Bryskin
Aihua Guo
ADVA Optical Networking

Expires: April 27, 2015

October 27, 2014

A Yang Data Model for Abstract TE Topologies
draft-liu-yang-abstract-te-topo-00

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), 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 April 27, 2015.

Abstract

This document discusses a YANG data model for Abstract TE Topologies.

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].

Table of Contents

1. Introduction

This document defines a YANG [RFC6020] [RFC6021] module for representing and manipulating Abstract TE topologies.

2. Abstract TE Topologies

2.1. Motivation

Clients of a transport network normally have no visibility into the network's actual Traffic-Engineering (TE) topology and resource availability information. There are numerous reasons for this, such as:

Security considerations: network operators are usually reluctant to expose the network's actual topology to its clients;

Transport network, generally speaking, is comprised of network elements that belong to a different layer network that the client devices. Also the internal network routing and traffic engineering advertisements usually contain proprietary information, which the clients cannot interpret, but discarding of which would lead to incorrect assumptions and decisions. This means that the clients cannot use actual network topology and traffic engineering information even if said information is available;

Scalability considerations: clients do not want to know any transport network information that is not related to the services provided to the clients.

On the other hand the clients need to influence to certain extent on the way the services provided to them are routed across the transport network: some services, for example, need to be as disjoint from each other as possible because they support various network failure protection schemes provisioned in the client layer network; others, on the contrary, need to be co-routed and share fate as much as possible; placement of some services needs to be optimized based on the lowest cost criteria, while other service paths need to be selected to have best optical signal quality or delay characteristics, and so forth.

Different approaches exist to allow for the clients to affect the placement of provided for them services on the transport network under conditions of no visibility into the actual transport network topology and resource availability information. For example, [GMPL-UNI] architecture allows for clients signaling their service routing policies/preferences within the service setup and modify messages and mandates the network path computers to honor said policies/preferences during the service path selection. There are also control plane based (e.g. [GMPLS-ENNI]) and SDN architectures that require the network to expose abstract TE topologies. Such topologies are decoupled from the network actual topologies and are provided on per client group/VPN/tenant basis. The abstract TE topologies are supposed to be fully comprehensible by the clients and contain sufficient information for the client path computers to select service paths according to the client policies. The service paths so selected in terms of abstract TE topology elements could be signaled or otherwise conveyed within service setup/modify requests to the transport network system responsible for the service provisioning.

2.2. Static vs Fluid Abstract TE Topologies

One problem with the abstract TE topologies exposed to the clients is their static nature. The abstract TE topologies are usually manually configured based on the transport network operator policies. This entails tedious error-prone configuration. This also does not allow for the clients to have a say as to how the abstract TE topologies exposed to them should look like, which elements (nodes, links) it should contain, what the parameters (e.g. link bandwidth, SRLGs, etc.) are, and so forth. The problem becomes especially profound in case the clients requirements with respect to the abstract TE topologies change over time and/or depend on particular week, day, time of the day, etc. It is highly desirable to have a data model understood and supported by the transport network and all its potential clients that would allow for the clients to dynamically (re-)configure the abstract TE topologies exposed to them in real time. This document introduces a data model written in YANG, that allows for the clients using NETCONF and/or RESTCONF protocols to (re-)configure abstract topologies, retrieve their data state and, thus, to automate the abstract topology manipulation.

3. Tree Structure

The structure of the groupings in this module are depicted below. Brackets enclose list keys, "rw" means configuration data, "ro" means operational state data, and "?" designates optional nodes.

```
module: abstract-te-topology
augment /nt:network-topology/nt:topology/nt:topology-types/l3t:l3-
unicast-igp-topology:
   +--rw abstract-te-topology!
augment /nt:network-topology/nt:topology/nt:node/nt:termination-
point/l3t:igp-termination-point-attributes:
   +--rw abstract-tp-attributes
      +--rw topo-ref?   leafref
      +--rw node-ref?   leafref
augment /nt:network-topology/nt:topology/nt:node/l3t:igp-node-
attributes:
   +--rw abstract-node-attributes
      +--rw schedules* [schedule-id]
      |  +--rw schedule-id          uint32
      |  +--rw start?               yang:date-and-time
```

```
         |  +--rw schedule-duration?   string
         |  +--rw repeat-interval?     string
         +--rw is-abstract?           boolean
         +--rw underlay-topology?     leafref
         +--rw connectivity-matrix* [id]
         |  +--rw id                          uint32
         |  +--rw from-tp
         |  |  +--rw topo-ref?   leafref
         |  |  +--rw node-ref?   leafref
         |  |  +--rw tp-ref?     leafref
         |  +--rw to-tp
         |  |  +--rw topo-ref?   leafref
         |  |  +--rw node-ref?   leafref
         |  |  +--rw tp-ref?     leafref
         |  +--rw is-allowed?               boolean
         |  +--rw information-source?       enumeration
         |  +--rw credibility-preference?   uint16
         +--rw ted
            +--rw te-router-id-ipv4?    inet:ipv4-address
            +--rw te-router-id-ipv6?    inet:ipv6-address
            +--rw ipv4-local-address* [ipv4-prefix]
            |  +--rw ipv4-prefix    inet:ipv4-prefix
            +--rw ipv6-local-address* [ipv6-prefix]
            |  +--rw ipv6-prefix      inet:ipv6-prefix
            |  +--rw prefix-option?   uint8
            +--rw pcc-capabilities?     pcc-capabilities
   augment /nt:network-topology/nt:topology/nt:link/l3t:igp-link-
   attributes:
      +--rw abstract-link-attributes
         +--rw schedules* [schedule-id]
         |  +--rw schedule-id          uint32
         |  +--rw start?               yang:date-and-time
         |  +--rw schedule-duration?   string
         |  +--rw repeat-interval?     string
         +--rw is-abstract?              boolean
         +--rw server-layer!
         |  +--rw dynamic?     boolean
         |  +--rw committed?   boolean
         +--rw server-path
         |  +--rw path-element* [path-element-id]
```

```
|     +--rw path-element-id    uint32
|     +--rw loose?             boolean
|     +--rw (element-type)?
|        +--:(numbered-link)
|        |  +--rw link-ip-address?   inet:ip-address
|        +--:(unnumbered-link)
|        |  +--rw link-node-id?      uint32
|        |  +--rw link-id?           uint32
|        +--:(node)
|        |  +--rw node-id?           uint32
|        +--:(label)
|           +--rw label?             uint32
+--rw server-backup-path
|  +--rw path-element* [path-element-id]
|     +--rw path-element-id    uint32
|     +--rw loose?             boolean
|     +--rw (element-type)?
|        +--:(numbered-link)
|        |  +--rw link-ip-address?   inet:ip-address
|        +--:(unnumbered-link)
|        |  +--rw link-node-id?      uint32
|        |  +--rw link-id?           uint32
|        +--:(node)
|        |  +--rw node-id?           uint32
|        +--:(label)
|           +--rw label?             uint32
+--rw server-protection-type?   uint16
+--rw server-trail-src
|  +--rw topo-ref?   leafref
|  +--rw node-ref?   leafref
|  +--rw tp-ref?     leafref
+--rw server-trail-des
|  +--rw topo-ref?   leafref
|  +--rw node-ref?   leafref
|  +--rw tp-ref?     leafref
+--rw ted
   +--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 color?                              uint32
            +--rw max-link-bandwidth?                 decimal64
            +--rw max-resv-link-bandwidth?            decimal64
            +--rw unreserved-bandwidth* [priority]
            |  +--rw priority     uint8
            |  +--rw bandwidth?   decimal64
            +--rw te-default-metric?                  uint32
            +--rw link-protection-type?               enumeration
            +--rw interface-switching-capabilities* [switching-
                                                       capability]
            |  +--rw switching-capability   ted:switching-capabilities
            |  +--rw encoding?              ted:encoding-type
            |  +--rw max-lsp-bandwidth* [priority]
            |  |  +--rw priority     uint8
            |  |  +--rw bandwidth?   decimal64
            |  +--rw packet-switch-capable
            |  |  +--rw minimum-lsp-bandwidth?   decimal64
            |  |  +--rw interface-mtu?           uint16
            |  +--rw time-division-multiplex-capable
            |     +--rw minimum-lsp-bandwidth?   decimal64
            |     +--rw indication?              enumeration
            +--rw srlg
               +--rw srlg-values* [srlg-value]
                  +--rw srlg-value    uint32
   augment /l3t:igp-node-event:
      +--ro abstract-te-topology!
      +--ro abstract-node-attributes
         +--ro schedules* [schedule-id]
         |  +--ro schedule-id          uint32
         |  +--ro start?               yang:date-and-time
         |  +--ro schedule-duration?   string
         |  +--ro repeat-interval?     string
         +--ro is-abstract?           boolean
         +--ro underlay-topology?     leafref
         +--ro connectivity-matrix* [id]
         |  +--ro id                        uint32
```

```
         |  +--ro from-tp
         |  |  +--ro topo-ref?   leafref
         |  |  +--ro node-ref?   leafref
         |  |  +--ro tp-ref?     leafref
         |  +--ro to-tp
         |  |  +--ro topo-ref?   leafref
         |  |  +--ro node-ref?   leafref
         |  |  +--ro tp-ref?     leafref
         |  +--ro is-allowed?               boolean
         |  +--ro information-source?       enumeration
         |  +--ro credibility-preference?   uint16
         +--ro ted
            +--ro te-router-id-ipv4?    inet:ipv4-address
            +--ro te-router-id-ipv6?    inet:ipv6-address
            +--ro ipv4-local-address* [ipv4-prefix]
            |  +--ro ipv4-prefix    inet:ipv4-prefix
            +--ro ipv6-local-address* [ipv6-prefix]
            |  +--ro ipv6-prefix      inet:ipv6-prefix
            |  +--ro prefix-option?   uint8
            +--ro pcc-capabilities?     pcc-capabilities
   augment /l3t:igp-link-event:
      +--ro abstract-te-topology!
      +--ro abstract-link-attributes
         +--ro schedules* [schedule-id]
         |  +--ro schedule-id          uint32
         |  +--ro start?               yang:date-and-time
         |  +--ro schedule-duration?   string
         |  +--ro repeat-interval?     string
         +--ro is-abstract?              boolean
         +--ro server-layer!
         |  +--ro dynamic?     boolean
         |  +--ro committed?   boolean
         +--ro server-path
         |  +--ro path-element* [path-element-id]
         |     +--ro path-element-id    uint32
         |     +--ro loose?             boolean
         |     +--ro (element-type)?
         |        +--:(numbered-link)
         |        |  +--ro link-ip-address?   inet:ip-address
         |        +--:(unnumbered-link)
```

```
|        |  +--ro link-node-id?      uint32
|        |  +--ro link-id?           uint32
|        +--:(node)
|        |  +--ro node-id?           uint32
|        +--:(label)
|           +--ro label?            uint32
+--ro server-backup-path
|  +--ro path-element* [path-element-id]
|     +--ro path-element-id    uint32
|     +--ro loose?             boolean
|     +--ro (element-type)?
|        +--:(numbered-link)
|        |  +--ro link-ip-address?   inet:ip-address
|        +--:(unnumbered-link)
|        |  +--ro link-node-id?      uint32
|        |  +--ro link-id?           uint32
|        +--:(node)
|        |  +--ro node-id?           uint32
|        +--:(label)
|           +--ro label?            uint32
+--ro server-protection-type?   uint16
+--ro server-trail-src
|  +--ro topo-ref?   leafref
|  +--ro node-ref?   leafref
|  +--ro tp-ref?     leafref
+--ro server-trail-des
|  +--ro topo-ref?   leafref
|  +--ro node-ref?   leafref
|  +--ro tp-ref?     leafref
+--ro ted
   +--ro link-index?                          uint64
   +--ro information-source?                  enumeration
   +--ro credibility-preference?              uint16
   +--ro admin-status?                        enumeration
   +--ro oper-status?                         enumeration
   +--ro area-id?                             binary
   +--ro color?                               uint32
   +--ro max-link-bandwidth?                  decimal64
   +--ro max-resv-link-bandwidth?             decimal64
   +--ro unreserved-bandwidth* [priority]
```

```
            |  +--ro priority     uint8
            |  +--ro bandwidth?   decimal64
            +--ro te-default-metric?                  uint32
            +--ro link-protection-type?               enumeration
            +--ro interface-switching-capabilities* [switching-
                                                       capability]
            |  +--ro switching-capability    ted:switching-capabilities
            |  +--ro encoding?               ted:encoding-type
            |  +--ro max-lsp-bandwidth* [priority]
            |  |  +--ro priority     uint8
            |  |  +--ro bandwidth?   decimal64
            |  +--ro packet-switch-capable
            |  |  +--ro minimum-lsp-bandwidth?   decimal64
            |  |  +--ro interface-mtu?           uint16
            |  +--ro time-division-multiplex-capable
            |     +--ro minimum-lsp-bandwidth?   decimal64
            |     +--ro indication?              enumeration
            +--ro srlg
               +--ro srlg-values* [srlg-value]
                  +--ro srlg-value    uint32
```

4. Abstract TE Topology - Yang Module

```
   module abstract-te-topology {
     yang-version 1;
     namespace "urn:ietf:params:xml:ns:yang:abstract-te-topology";
     // replace with IANA namespace when assigned

     prefix "abst";

     import ietf-yang-types {
       prefix "yang";
     }

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

     import network-topology {
       prefix "nt";
```

```
  }

  import l3-unicast-igp-topology {
    prefix "l3t";
  }

  import ted {
    prefix "ted";
  }

  organization "TBD";
  contact "TBD";
  description "Abstract topology model";

  revision "2014-10-27" {
    description "Initial revision";
    reference "TBD";
  }

  grouping abstract-te-topology-type {
    description
      "Identifies the abstract topology type.";
    container abstract-te-topology {
      presence "indicates abstract topology";
      description
        "Its presence identifies the abstract topology type.";
    }
  }

  augment "/nt:network-topology/nt:topology/"
          + "nt:topology-types/l3t:l3-unicast-igp-topology" {
    description
      "Defines the abstract topology type.";
    uses abstract-te-topology-type;
  }

  grouping te-path-element {
    description
      "A group of attributes defining an element in a TE path
      such as TE node, TE link, TE aotomic resource or label.";
    leaf loose {
      type boolean;
      description "true if the element is loose.";
    }
    choice element-type {
      description "Attributes for various element types.";
```

```
        case numbered-link {
          leaf link-ip-address {
            type inet:ip-address;
            description "IPv4 or IPv6 address.";
          }
        }
        case unnumbered-link {
          leaf link-node-id {
            type uint32;
            description
              "Node ID of the node where the link end point resides.";
          }
          leaf link-id {
            type uint32;
            description "Identifies the link end point.";
          }
        }
        case node {
          leaf node-id {
            type uint32;
            description "Identifies the node.";
          }
        }
        case label {
          leaf label {
            type uint32;
            description "Identifies atomic TE resource or label.";
          }
        }
      }
    } // te-path-element

    grouping config-schedule-attributes {
      description
        "A list of schedules defining when a particular
         configuration takes effect.";
      list schedules {
        key "schedule-id";
        description "A list of schedule elements.";

        leaf schedule-id {
          type uint32;
          description "Identifies the schedule element.";
        }
        leaf start {
          type yang:date-and-time;
```

```
          description "Start time.";
        }
        leaf schedule-duration {
          type string {
            pattern
              'P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)?';
          }
          description "Schedule duration in ISO 8601 format.";
        }
        leaf repeat-interval {
          type string {
            pattern
              'R\d*/P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?'
              + '(\d+S)?';
          }
          description "Repeat interval in ISO 8601 format.";
        }
      }
    }

    grouping abstract-node-attributes {
      description "Node attributes in an abstract topology.";
      container abstract-node-attributes {
        description "Node attributes in an abstract topology.";
        uses config-schedule-attributes;
        leaf is-abstract {
          type boolean;
          description
            "true if the node is abstract, false when the node is
            actual.";
        }
        leaf underlay-topology {
          type leafref {
            path "/nt:network-topology/nt:topology/nt:topology-id";
          }
          description
            "When an abstract node encapsulates a topology,
             this reference points to said topology.";
        }
        list connectivity-matrix {
          key "id";
          description
            "Represents node's switching limitations, i.e. limitations
             in interconnecting network termination points (NTPs)
             across the node.";
          leaf id {
```

```
            type uint32;
            description "Identifies the connectivity-matrix entry.";
          }
          container from-tp {
            uses l3t:tp-ref;
            description
              "Reference to source NTP.";
          }
          container to-tp {
            uses l3t:tp-ref;
            description
              "Reference to destination NTP.";
          }
          leaf is-allowed {
            type boolean;
            description
              "true  - switching is allowed,
               false - switching is disallowed.";
          }
          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";
              }
              enum "other" {
                description "Other source";
              }
            }
            description
              "Indicates the source of the information.";
          }
          leaf credibility-preference {
            type uint16;
            description
```

```
              "The preference value to calculate the traffic
               engineering database credibility value used for
               tie-break selection between different
               information-source values.
               Higher value is more preferable.";
          }

        }
        container ted {
          description "Includes TE node attributes.";
          uses ted:ted-node-attributes;
        }
      }
    } // abstract-node-attributes

    grouping abstract-tp-attributes {
      description
        "Termination point attributes in an abstract topology.";
      container abstract-tp-attributes {
        description
          "Termination point attributes in an abstract topology.";
        uses l3t:node-ref;
      }
    } // abstract-tp-attributes

    grouping abstract-link-attributes {
      description
        "Link attributes in an abstract topology.";
      container abstract-link-attributes {
        description "Link attributes in an abstract topology.";
        uses config-schedule-attributes;
        leaf is-abstract {
          type boolean;
          description "true if the link is abstract.";
        }
        container server-layer {
          presence
            "Indicates the server layer exists for this link.";
          description "State of the server layer of this link.";

          leaf dynamic {
            type boolean;
            description
              "true if the server layer is dynamically created.";
          }
          leaf committed {
```

```
            type boolean;
            description
              "true if the server layer is committed.";
          }
        }
        container server-path {
          description
            "The service path on the server layer topology that
             supports this link.";
          list path-element {
            key "path-element-id";
            description
              "A list of path elements describing the service path";
            leaf path-element-id {
              type uint32;
              description "To identify the element in a path.";
            }
            uses te-path-element;
          }
        } // server-path
        container server-backup-path {
          description
            "The backup service path on the server layer topology that
             supports this link.";
          list path-element {
            key "path-element-id";
            description
              "A list of path elements describing the backup service
               path";
            leaf path-element-id {
              type uint32;
              description "To identify the element in a path.";
            }
            uses te-path-element;
          }
        } // server-backup-path
        leaf server-protection-type {
          type uint16;
          description
            "Server layer protection type desired for this link";
        }
        container server-trail-src {
          uses l3t:tp-ref;
          description
            "Source termination point of the server layer trail.";
        }
```

```
        container server-trail-des {
          uses l3t:tp-ref;
          description
            "Destination termination point of the server layer
             trail.";
        }
        container ted {
          description "Includes TE link attributes.";
          uses ted:ted-link-attributes;
        }
      }
    } // abstract-link-attributes

    augment "/nt:network-topology/nt:topology/nt:node/"
            + "nt:termination-point/"
            + "l3t:igp-termination-point-attributes" {
      when "../../../topology-types/abstract-te-topology" {
        description
          "The augment is valid only for abstract topology.";
      }
      description "Augments attributes on a termination point.";
      uses abstract-tp-attributes;
    }

    augment "/nt:network-topology/nt:topology/nt:node/"
            + "l3t:igp-node-attributes" {
      when "../../topology-types/abstract-te-topology" {
        description
          "The augment is valid only for abstract topology.";
      }
      description "Augments attributes on a node.";
      uses abstract-node-attributes;
    }

    augment "/nt:network-topology/nt:topology/nt:link/"
            + "l3t:igp-link-attributes" {
      when "../../topology-types/abstract-te-topology" {
        description
          "The augment is valid only for abstract topology.";
      }
      description "Augments attributes on a link.";
      uses abstract-link-attributes;
    }

    augment "/l3t:igp-node-event" {
      description "Augments node event.";
```

```
      uses abstract-te-topology-type;
      uses abst:abstract-node-attributes;
    }

    augment "/l3t:igp-link-event" {
      description "Augments link event.";
      uses abstract-te-topology-type;
      uses abst:abstract-link-attributes;
    }
  }
```

5. Security Considerations

The protocol used for sending the TE topology data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications.

6. IANA Considerations

TBD

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", RFC 6021, October 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, Internet Mail Consortium and Demon Internet Ltd., November 1997.

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[RFC3811] Nadeau, T. and J. Cucchiara, "Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management", RFC 3811, June 2004.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

[RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)", RFC 3813, June 2004.

[RFC4208] Swallow, G., Drake, J., Ishimatsu, H., and Rekhter, Y., "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model", RFC4208, October 2005.

[RFC4220] Dubuc, M., Nadeau, T., and Lang, J., " Traffic Engineering Link Management Information Base", RFC 4220, November 2005.

[RFC4801] Nadeau, T., Ed. and A. Farrel, Ed., "Definitions of Textual Conventions for Multiprotocol Label Switching (MPLS) Management", RFC 4801, February 2007.

[RFC4802] Nadeau, T., Ed. and A. Farrel, Ed., "Generalized Multiprotocol Label Switching (GMPLS) Traffic Engineering Management Information Base", RFC 4802, February 2007.

7.2. Informative References

[G.8080] ITU-T Rec. G.8080/Y.1304, "Architecture for the Automatically Switched Optical Network (ASON)," November 2001 (and Revision, January 2003). For information on the availability of this document, please see http://www.itu.int.

[I-D.clemm-i2rs-yang-network-topo]

A. Clemm, "A YANG Data Model for Network Topologies", draft-clemm-i2rs-yang-network-topo-01.

[I-D.liu-yang-ted]

Xufeng Liu, "A Yang module for TED", draft-liu-yang-ted-00.

[I-D.clemm-i2rs-yang-l3-topo]

A. Clemm,"A YANG Data Model for Layer 3 Topologies", draft-clemm-i2rs-yang-l3-topo-00.

8. Acknowledgments

TBD

Authors' Addresses

Xufeng Liu
Ericsson
Email: Xufeng.liu@ericsson.com

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Alexander Clemm
Cisco
Email: alex@cisco.com

Igor Bryskin
ADVA Optical Networking
Email: ibryskin@advaoptical.com

Aihua Guo
ADVA Optical Networking
Email: aguo@advaoptical.com

Contributors

Gert Grammel
Juniper Networks
Email: ggrammel@juniper.net