

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 2015

X. Liu
Ericsson
I. Bryskin
ADVA Optical Networking
A. Clemm
Cisco
V. P. Beeram
Juniper Networks
July 1, 2014

A YANG Data Model for Abstract Network Topologies
draft-liu-netmod-yang-abstract-topo-00.txt

Abstract

This document describes a concept, a methodology and a YANG data model to (re-)configure abstract topologies, retrieve their states and thus to automate the abstract topology manipulation.

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 January 1, 2015.

Copyright Notice

Copyright (c) 2014 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.

Table of Contents

1.	Introduction.....	2
1.1.	Terminology.....	4
2.	Abstract network topology model overview.....	4
3.	Usage Example.....	9
4.	Abstract Network Topology YANG module.....	24
5.	Security Considerations.....	32
6.	References.....	32
6.1.	Normative References.....	32
6.2.	Informative References.....	33

[1.](#) Introduction

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

- a) Security considerations: network operators are usually reluctant to expose the network's actual topology to its clients;
- b) Transport network, generally speaking, is comprised of network elements that belong to a different layer network than 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;
- c) 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 topologies. Such topologies are decoupled from the network actual topologies and are provided on per client group/VPN/tenant basis. The abstract topologies are supposed to be fully understandable 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 topology elements could be signaled or otherwise conveyed within service setup/modify requests to the transport network system responsible for the service provisioning.

One problem with the abstract topologies exposed to the clients is their static nature. The abstract 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 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 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 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.

1.1. Terminology

The keywords "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](#)].

The following terms are defined in [[RFC6020](#)] and are not redefined here:

- o augment
- o data model
- o data node

2. Abstract network topology model overview

This document defines the YANG module "abstract-topology", which has the following structure:

```
module: abstract-topology
augment /nt:network-topology/nt:topology/nt:topology-types/l3t:l3-
unicast-igp-topology:
  +--rw abstract!
augment /nt:network-topology/nt:topology/nt:node/nt:termination-
point/l3t:igp-termination-point-attributes:
  +--rw abstract-tp-attributes
    +--rw node-ref?   nt:node-ref
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-virtual?        boolean
    +--rw underlay-topology?  nt:topology-ref
    +--rw connectivity-matrix* [from-tp to-tp]
      | +--rw from-tp         nt:tp-ref
```

```
| +--rw to-tp      nt:tp-ref
+--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 advertising-node-src?  nt:node-id
  +--rw advertising-node-des?  nt:node-id
  +--rw is-virtual?            boolean
  +--ro melg*                  uint32
  +--ro srclg*                 uint32
  +--rw server-path
  | +--rw path-element* [path-element-id]
  |   +--rw path-element-id  uint32
  |   +--rw loose?          boolean
  |   +--rw (element-type)?
  |     +--:(numbered-link)
  |       | +--rw link-ipv4?      uint32
  |       +--:(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-ipv4?          uint32
|      +---:(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?        nt:tp-ref
+--rw server-trail-des?        nt:tp-ref
+--rw ted
|   +--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 srlg
|   +--rw interface-switching-capabilities* [switching-
capability]
|   |   +--rw switching-capability
ted:switching-capabilities
|   |   +--rw encoding?          uint8
|   |   +--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?     uint16
|   +--rw srlg-values* [srlg-value]
|   |   +--rw srlg-value        uint32
+--rw link-protection-type?      uint16
augment /l3t:igp-node-event:
  +--ro abstract!

```

```
+--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-virtual?         boolean
  +--ro underlay-topology?  nt:topology-ref
  +--ro connectivity-matrix* [from-tp to-tp]
    | +--ro from-tp      nt:tp-ref
    | +--ro to-tp       nt:tp-ref
  +--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!
  +--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 advertising-node-src?  nt:node-id
    +--ro advertising-node-des?  nt:node-id
    +--ro is-virtual?            boolean
    +--ro melg*                  uint32
    +--ro srclg*                 uint32
    +--ro server-path
      | +--ro path-element* [path-element-id]
      |   +--ro path-element-id  uint32
      |   +--ro loose?          boolean
      |   +--ro (element-type)?
      |     +--:(numbered-link)
      |       | +--ro link-ipv4?      uint32
      |       +--:(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-ipv4?         uint32
|       +---:(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?          nt:tp-ref
+--ro server-trail-des?          nt:tp-ref
+--ro ted
  +--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 srlg
    +--ro interface-switching-capabilities* [switching-
capability]
      | +--ro switching-capability
ted:switching-capabilities
      | +--ro encoding?                               uint8
      | +--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?            uint16
+--ro srlg-values* [srlg-value]
| +--ro srlg-value    uint32
+--ro link-protection-type?          uint16

```

3. Usage Example

Figure 1 shows an example of abstract network topology. This topology consists of four nodes at the physical layer: E, F, H, and H, which are connected by four physical links: <E, F>, <F, G>, <G, H>, and <E, H>. Nodes E and F are grouped into a virtual node VN1; nodes G and H are grouped into a virtual node VN2. There is a virtual link from node E in VN1 to node G in VN2.

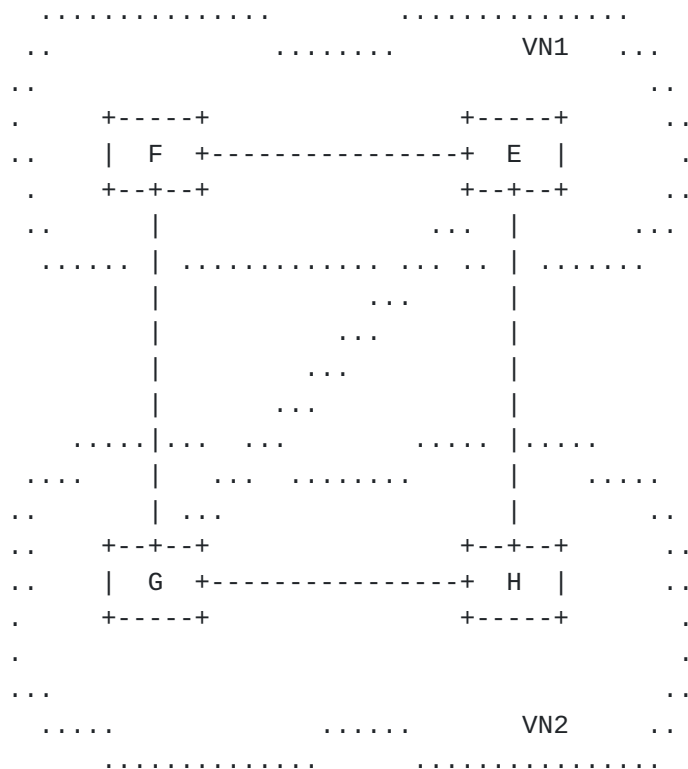


Figure 1 Example of Abstract Network Topology

The JSON encoded configuration example for such a topology can be as following:

```
{
  "network-topology": {
    "topology": [
      {
        "topology-id": "ODUK",
        "topology-types": {
          "l3-unicast-igp-topology": {
            "abstract": {}
          }
        },
        "node": [
          {
            "node-id": "VN1",
            "termination-point": [
              {
                "tp-id": "E#CH-1-5-NE",
                "igp-termination-point-attributes": {
                  "unnumbered-id": 101,
                  "abstract-tp-attributes": {
                    "node-ref": "E"
                  }
                }
              }
            ],
            "igp-node-attributes": {
              "name": "VN1",
              "abstract-node-attributes": {
                "is-virtual": true,
                "underlay-topology": "/network-
topology/topology[topology-id='VN1']"
              }
            }
          }, // VN1
          {
            "node-id": "VN2",
            "termination-point": [
              {
                "tp-id": "G#CH-1-6-NE",
```

```
        "igp-termination-point-attributes": {
          "unnumbered-id": 102,
          "abstract-tp-attributes": {
            "node-ref": "G"
          }
        }
      },
    ],
    "igp-node-attributes": {
      "name": "VN2",
      "abstract-node-attributes": {
        "is-virtual": true,
        "underlay-topology": "/network-
topology/topology[topology-id='VN2']"
      }
    }
  } // VN2
],
"link": [
  {
    "link-id": "VN1#101-VN2#102",
    "source": {
      "source-node": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN1']",
      "source-tp": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN1']/termination-point[tp-id='E#CH-1-5-
NE']"
    }
    "destination": {
      "dest-node": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN2']",
      "dest-tp": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN2']/termination-point[tp-id='G#CH-1-6-
NE']"
    }
    "supporting-link": [],
    "igp-link-attributes": {
      "abstract-link-attributes": {
        "srlg": {
          "interface-switching-capabilities": [
            {
```

```

    "switching-capability": "OTN-TDM", // 110,
RFC7138
    "encoding": 12, // G.709 OKUK, RFC4328
    "max-lsp-bandwidth": [
        {
            "priority": 7,
            "bandwidth": 1254659200.0 // ODU2 RFC7138
        }
    ] // max-lsp-bandwidth
    }
  ] // interface-switching-capabilities
}
"is-virtual": true,
"advertising-node-src": "E",
"advertising-node-des": "G"
} // abstract-link-attributes
} // igp-link-attributes
}, // link "VN1#101-VN2#102"
{
  "link-id": "VN2#102-VN1#101",
  "source": {
    "source-node": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN2']",
    "source-tp": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN2']/termination-point[tp-id='G#CH-1-6-
NE']"
  }
  "destination": {
    "dest-node": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN1']",
    "dest-tp": "/network-topology/topology[topology-
id='ODUK']/node[node-id='VN1']/termination-point[tp-id='E#CH-1-5-
NE']"
  }
  "supporting-link": [],
  "igp-link-attributes": {
    "abstract-link-attributes": {
      "srlg": {
        "interface-switching-capabilities": [
          {

```

[RFC7138](#)

```
        "switching-capability": "OTN-TDM", // 110,
        "encoding": 12, // G.709 OKUK, RFC4328
        "max-lsp-bandwidth": [
            {
                "priority": 7,
                "bandwidth": 1254659200.0 // ODU2 RFC7138
            }
        ] // max-lsp-bandwidth
    }
    ] // interface-switching-capabilities
}
"is-virtual": true,
"advertising-node-src": "G",
"advertising-node-des": "E"
} // abstract-link-attributes
} // igp-link-attributes
} // "VN2#102-VN1#101"
] // link
}, // topoloty "ODUK"
{
    "topology-id": "WDM",
    "topology-types": {
        "l3-unicast-igp-topology": {
            "abstract": {}
        }
    },
    "node": [
        {
            "node-id": "E",
            "termination-point": [
                {
                    "tp-id": "OL-1",
                    "igp-termination-point-attributes": {
                        "unnumbered-id": 101
                    }
                },
                {
                    "tp-id": "OL-2",
                    "igp-termination-point-attributes": {
                        "unnumbered-id": 102
                    }
                }
            ]
        }
    ]
}
```

```
    }
  },
  {
    "tp-id": "CH-1-5-NE",
    "igp-termination-point-attributes": {
      "unnumbered-id": 103
    }
  }
],
"igp-node-attributes": {
  "name": "E",
  "abstract-node-attributes": {
    "is-virtual": false
  }
}
}, // node E
{
  "node-id": "F",
  "termination-point": [
    {
      "tp-id": "OL-1",
      "igp-termination-point-attributes": {
        "unnumbered-id": 101
      }
    },
    {
      "tp-id": "OL-2",
      "igp-termination-point-attributes": {
        "unnumbered-id": 102
      }
    }
  ],
  "igp-node-attributes": {
    "name": "F",
    "abstract-node-attributes": {
      "is-virtual": false
    }
  }
}
}, // node F
{
  "node-id": "G",
```

```
"termination-point": [  
  {  
    "tp-id": "OL-1",  
    "igp-termination-point-attributes": {  
      "unnumbered-id": 101  
    }  
  },  
  {  
    "tp-id": "OL-2",  
    "igp-termination-point-attributes": {  
      "unnumbered-id": 102  
    }  
  },  
  {  
    "tp-id": "CH-1-6-NE",  
    "igp-termination-point-attributes": {  
      "unnumbered-id": 103  
    }  
  }  
],  
"igp-node-attributes": {  
  "name": "G",  
  "abstract-node-attributes": {  
    "is-virtual": false  
  }  
}  
}, // node G  
{  
  "node-id": "H",  
  "termination-point": [  
    {  
      "tp-id": "OL-1",  
      "igp-termination-point-attributes": {  
        "unnumbered-id": 101  
      }  
    },  
    {  
      "tp-id": "OL-2",  
      "igp-termination-point-attributes": {  
        "unnumbered-id": 102  
      }  
    }  
  ]  
}
```

```
    }
  ],
  "igp-node-attributes": {
    "name": "H",
    "abstract-node-attributes": {
      "is-virtual": false
    }
  }, // node H
],
"link": [
  {
    "link-id": "E#101-F#102",
    "source": {
      "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']",
      "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']/termination-point[tp-id='OL-1']"
    }
    "destination": {
      "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']",
      "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']/termination-point[tp-id='OL-2']"
    }
    "supporting-link": [],
    "igp-link-attributes": {
      "abstract-link-attributes": {
        "srlg": {
          "interface-switching-capabilities": [
            {
              "switching-capability": "LSC",
              "encoding": 12, // G.709 OKUK, RFC4328
              "max-lsp-bandwidth": [
                {
                  "priority": 7,
                  "bandwidth": 1254659200.0 // ODU2 RFC7138
                }
              ] // max-lsp-bandwidth
            }
          ] // interface-switching-capabilities
        }
      }
    }
  }
]
```

```
    }
    "is-virtual": false
  } // abstract-link-attributes
} // igp-link-attributes
}, // link E-F
{
  "link-id": "F#102-E#101",
  "source": {
    "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']",
    "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']/termination-point[tp-id='OL-2']"
  }
  "destination": {
    "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']",
    "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']/termination-point[tp-id='OL-1']"
  }
  "supporting-link": [],
  "igp-link-attributes": {
    "abstract-link-attributes": {
      "srlg": {
        "interface-switching-capabilities": [
          {
            "switching-capability": "LSC",
            "encoding": 12, // G.709 OKUk, RFC4328
            "max-lsp-bandwidth": [
              {
                "priority": 7,
                "bandwidth": 1254659200.0 // ODU2 RFC7138
              }
            ] // max-lsp-bandwidth
          }
        ] // interface-switching-capabilities
      }
    }
    "is-virtual": false
  } // abstract-link-attributes
} // igp-link-attributes
}, // link F-E
{
```

```
    "link-id": "E#102-H#101",
    "source": {
      "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']",
      "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']/termination-point[tp-id='OL-2']"
    }
    "destination": {
      "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']",
      "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']/termination-point[tp-id='OL-1']"
    }
    "supporting-link": [],
    "igp-link-attributes": {
      "abstract-link-attributes": {
        "srlg": {
          "interface-switching-capabilities": [
            {
              "switching-capability": "LSC",
              "encoding": 12, // G.709 OKUK, RFC4328
              "max-lsp-bandwidth": [
                {
                  "priority": 7,
                  "bandwidth": 1254659200.0 // ODU2 RFC7138
                }
              ] // max-lsp-bandwidth
            }
          ] // interface-switching-capabilities
        }
        "is-virtual": false
      } // abstract-link-attributes
    } // igp-link-attributes
  }, // link E-H
  {
    "link-id": "H#101-E#102",
    "source": {
      "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']",
      "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']/termination-point[tp-id='OL-1']"
```

```
    }
    "destination": {
      "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']",
      "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='E']/termination-point[tp-id='OL-2']"
    }
    "supporting-link": [],
    "igp-link-attributes": {
      "abstract-link-attributes": {
        "srlg": {
          "interface-switching-capabilities": [
            {
              "switching-capability": "LSC",
              "encoding": 12, // G.709 OKUk, RFC4328
              "max-lsp-bandwidth": [
                {
                  "priority": 7,
                  "bandwidth": 1254659200.0 // ODU2 RFC7138
                }
              ] // max-lsp-bandwidth
            }
          ] // interface-switching-capabilities
        }
        "is-virtual": false
      } // abstract-link-attributes
    } // igp-link-attributes
  }, // link H-E
  {
    "link-id": "F#101-G#102",
    "source": {
      "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']",
      "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']/termination-point[tp-id='OL-1']"
    }
    "destination": {
      "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']",
      "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']/termination-point[tp-id='OL-2']"
```

```
}
"supporting-link": [],
"igp-link-attributes": {
  "abstract-link-attributes": {
    "srlg": {
      "interface-switching-capabilities": [
        {
          "switching-capability": "LSC",
          "encoding": 12, // G.709 OKUk, RFC4328
          "max-lsp-bandwidth": [
            {
              "priority": 7,
              "bandwidth": 1254659200.0 // ODU2 RFC7138
            }
          ] // max-lsp-bandwidth
        }
      ] // interface-switching-capabilities
    }
    "is-virtual": false
  } // abstract-link-attributes
} // igp-link-attributes
}, // link F-G
{
  "link-id": "G#102-F#101",
  "source": {
    "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']",
    "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']/termination-point[tp-id='OL-2']"
  }
  "destination": {
    "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']",
    "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='F']/termination-point[tp-id='OL-1']"
  }
  "supporting-link": [],
  "igp-link-attributes": {
    "abstract-link-attributes": {
      "srlg": {
        "interface-switching-capabilities": [
```

```
    {
      "switching-capability": "LSC",
      "encoding": 12, // G.709 OKUK, RFC4328
      "max-lsp-bandwidth": [
        {
          "priority": 7,
          "bandwidth": 1254659200.0 // ODU2 RFC7138
        }
      ] // max-lsp-bandwidth
    }
  ] // interface-switching-capabilities
}
"is-virtual": false
} // abstract-link-attributes
} // igp-link-attributes
}, // link G-F
{
  "link-id": "G#101-H#102",
  "source": {
    "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']",
    "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']/termination-point[tp-id='OL-1']"
  }
  "destination": {
    "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']",
    "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']/termination-point[tp-id='OL-2']"
  }
  "supporting-link": [],
  "igp-link-attributes": {
    "abstract-link-attributes": {
      "srlg": {
        "interface-switching-capabilities": [
          {
            "switching-capability": "LSC",
            "encoding": 12, // G.709 OKUK, RFC4328
            "max-lsp-bandwidth": [
              {
                "priority": 7,
```

```
        "bandwidth": 1254659200.0 // ODU2 RFC7138
      }
    ] // max-lsp-bandwidth
  }
] // interface-switching-capabilities
}
"is-virtual": false
} // abstract-link-attributes
} // igp-link-attributes
}, // link G-H
{
  "link-id": "H#102-G#101",
  "source": {
    "source-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']",
    "source-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='H']/termination-point[tp-id='OL-2']"
  }
  "destination": {
    "dest-node": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']",
    "dest-tp": "/network-topology/topology[topology-
id='WDM']/node[node-id='G']/termination-point[tp-id='OL-1']"
  }
  "supporting-link": [],
  "igp-link-attributes": {
    "abstract-link-attributes": {
      "srlg": {
        "interface-switching-capabilities": [
          {
            "switching-capability": "LSC",
            "encoding": 12, // G.709 OKUk, RFC4328
            "max-lsp-bandwidth": [
              {
                "priority": 7,
                "bandwidth": 1254659200.0 // ODU2 RFC7138
              }
            ] // max-lsp-bandwidth
          }
        ] // interface-switching-capabilities
      }
    }
  }
}
```

```
        "is-virtual": false
      } // abstract-link-attributes
    } // igp-link-attributes
  } // link H-G
],
}, // topology "WDM"
{
  "topology-id": "VN1",
  "topology-types": {
    "l3-unicast-igp-topology": {
      "abstract": {}
    }
  },
  "node": [
    {
      "node-id": "E",
      "supporting-node": [
        {
          "node-ref": "E"
        }
      ]
    }, // ref to E
    {
      "node-id": "F",
      "supporting-node": [
        {
          "node-ref": "F"
        }
      ]
    } // ref to F
  ]
}, // topoloty "VN1"
{
  "topology-id": "VN2",
  "topology-types": {
    "l3-unicast-igp-topology": {
      "abstract": {}
    }
  },
  "node": [
    {
```

```
        "node-id": "G",
        "supporting-node": [
            {
                "node-ref": "G"
            }
        ], // ref to G
    {
        "node-id": "H",
        "supporting-node": [
            {
                "node-ref": "H"
            }
        ], // ref to H
    } // topoloty "VN2"
]
} // network-topology
}
```

4. Abstract Network Topology YANG module

<CODE BEGINS> file "abstract-topology@2014-07-01.yang"

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

    prefix "abst";

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

    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 "2014-07-01" {
    description "Initial revision";
    reference "TBD";
  }

  grouping abstract-topology-type {
    description
      "Identifies the abstract topology type.";
    container abstract {
      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-topology-type;
  }

  grouping te-path-element {
    description
      "A group of attributes defining an element in a TE path";
    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-ipv4 {
    type uint32;
    description "IPv4 address in 4 byte integer format.";
  }
}
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 the 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-virtual {
      type boolean;
      description "true if the node is virtual.";
    }
    leaf underlay-topology {
      type nt:topology-ref;
      description
        "When a node contains a topology, such as a virtual node,
        this reference points to the topology that defines the
        topology inside this node.";
    }
  }
}
```

```
    }
    list connectivity-matrix {
      key "from-tp to-tp";
      description
        "Representation of the limit to the connectivity within
        the node";
      leaf from-tp {
        type nt:tp-ref;
        description
          "Reference to source connectivity point.";
      }
      leaf to-tp {
        type nt:tp-ref;
        description
          "Reference to destination connectivity point.";
      }
    }
    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.";
    leaf node-ref {
      type nt:node-ref;
      description "Node where this termination point resides.";
    }
  }
} // 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 advertising-node-src {
    type nt:node-id;
    description
        "The node that advertises the source link end point";
}
leaf advertising-node-des {
    type nt:node-id;
    description
        "The node that advertises the destination link end point";
}
leaf is-virtual {
    type boolean;
    description "true if the link is virtual.";
}
leaf-list melg {
    type uint32;
    config false;
    description "A list of MELG values of the link.";
}
leaf-list srclg {
    type uint32;
    config false;
    description "A list of SRCLG values of the link.";
}
container server-path {
    description
        "The service path on the server layer 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 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";
  }
  leaf server-trail-src {
    type nt:tp-ref;
    description
      "Source termination point of the server layer trail.";
  }
  leaf server-trail-des {
    type nt: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-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-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-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-topology-type;
    uses abst:abstract-node-attributes;
  }

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

<CODE ENDS>

5. Security Considerations

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

6. References

6.1. Normative References

- [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.

[6.2. 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-netmod-yang-network-topo]
- A. Clemm, H. Ananthakrishnan, J. Medved, T. Tkacik, R. Varga, and N. Bahadur, "A YANG Data Model for Network Topologies", [draft-clemm-i2rs-yang-network-topo-00](#)
- [I-D.beeram-ccamp-melg]
- V. P. Beeram, I. Bryskin, "Mutually Exclusive Link Group (MELG)", [draft-beeram-ccamp-melg-03](#)
- [I-D.beeram-ccamp-srclg]
- V. P. Beeram, I. Bryskin, "Shared Resource Link Group (SRCLG)", [draft-beeram-ccamp-srclg-01](#)

Authors' Addresses

Xufeng Liu
Ericsson

Email: xufeng.liu@ericsson.com

Igor Bryskin
ADVA Optical Networking

Email: ibryskin@advaoptical.com

Alexander Clemm
Cisco

Email: alex@cisco.com

Vishnu Pavan Beeram
Juniper Networks

Email: vbeeram@juniper.net