Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 30, 2015

A. Clemm
J. Medved
Cisco
R. Varga
T. Tkacik
Pantheon Technologies SRO
X. Liu
Ericsson
I. Bryskin
A. Guo
Adva Optical
H. Ananthakrishnan
Packet Design
N. Bahadur
Bracket Computing
V. Beeram
Juniper Networks

October 27, 2014

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

Abstract

This document defines a YANG data model for layer 3 network topologies.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document introduces a YANG [RFC6020] [RFC6021] data model for Layer 3 network topologies. The model allows an application to have a holistic view of the topology of a Layer 3 network, all contained in a single conceptual YANG datastore. The data model builds on top of, and augments, the data model for network topologies defined in [topology-dm]. An earlier revision of that Internet Draft contained not just the general model for network topologies, but also the model for layer 3 network topologies that is being specified here. However, we decided to "split" the earlier draft to separate the truly general aspects of a topology data model, which apply to any type of topology, from the application of this model to a particular domain, here: a Layer 3 network.

Specific topology types that are covered in this document include Layer 3 Unicast IGP, IS-IS [RFC1195], and OSPF [RFC2178]. In addition, this documents defines a set of traffic engineering extensions.

There are multiple applications for such a data model. For example, nodes within the network can use the data model to capture their understanding of the overall network topology and expose it to a network controller. A network controller can then use the instantiated topology data to compare and reconcile its own view of the network topology with that of the network elements that it controls. Alternatively, nodes within the network could propagate this understanding to compare and reconcile this understanding either amongst themselves or with help of a controller. Beyond the network element itself, a network controller might even use the data model to represent its view of the topology that it controls and expose it to applications north of itself.

There are several reasons to choose YANG to define the data model. Data defined using YANG can be exposed by a server to client applications and controllers via Netconf [RFC6241] or via a ReST Interface [restconf] [yang-json]. The fact that it can be used with different protocols and interfaces provides for a degree of "future-proofing" of model implementations. Also, YANG can serve as the basis for model-driven toolchains, such as used in the Open Daylight project.

The data model is defined in several YANG modules:

- Module "l3-unicast-igp-topology" defines a model for Layer 3 Unicast IGP topologies. To do so, it augments general network topology model defined in [topology-dm] with information specific to Layer 3 Unicast IGP. In doing so, it also illustrates the extension patterns associated with extending respectively augmenting the general topology model to meet the needs of a specific topology.

- Module "ospf-topology" defines a topology model for OSPF, building on and extending the Layer 3 Unicast IGP topology model. It serves as an example of how the general topology model can be refined across multiple levels.

- Module "isis-topology" defines a topology model for IS-IS, again building on and extending the Layer 3 Unicast IGP topology model.

- Module "ted", finally, is a helper module, defining information kept in the Traffic Engineering Database (TED) that is leveraged by IS-IS and OSPF topologies. EDITOR’S NOTE: This module will be removed in subsequent revisions. Instead, it will be specified in a separate draft, [yang-ted].

2. Definitions and Acronyms

Datastore: A conceptual store of instantiated management information, with individual data items represented by data nodes which are arranged in hierarchical manner.

Data subtree: An instantiated data node and the data nodes that are hierarchically contained within it.

HTTP: Hyper-Text Transfer Protocol

IGP: Interior Gateway Protocol

IS-IS: Intermediate System to Intermediate System protocol

LSP: Label Switched Path

NETCONF: Network Configuration Protocol

OSPF: Open Shortest Path First, a link state routing protocol

URI: Uniform Resource Identifier

ReST: Representational State Transfer, a style of stateless interface and protocol that is generally carried over HTTP

SRLG: Shared Risk Link Group

TED: Traffic Engineering Database

YANG: A data definition language for NETCONF

3. Model overview

This section provides an overview of the Layer 3 network topology model.

3.1. Model structure

The network topology model is defined by the following YANG modules, whose relationship is roughly depicted in the figure below. The base network topology is included in the diagram for completeness.

```
              +-----------------------+
              |    network-topology   |
              |      [topology-dm]    |
              +-----------+-----------+
                          ^
                          |
                          |
              +-----------^-------------+
              | l3-unicast-igp-topology |
              +----+---------------+----+
                   ^               ^
                   |               |
                   |               |
          +--------^-----+   +-----^---------+    +--------+
          | ospf-topology|   | isis-topology |    |  ted   |
          +--------^-----+   +-----^---------+    +----v---+
                   :               :                   :
                   :...............:...................:
```

Figure 1: Overall model structure

YANG module network-topology defines the basic network topology model. YANG module l3-unicast-igp-topology augments network-topology with additional definitions needed to represent Layer 3 Unicast IGP topologies. This module in turn is augmented by YANG modules with additional definitions for OSPF and for IS-IS topologies, ospf-topology and isis-topology, respectively. Finally, YANG module "ted" contains a set of auxiliary definitions used by both ospf-topology and isis-topology, capturing data related to traffic engineering. EDITOR’S NOTE: Module "ted" will be removed from subsequent revisions and be specified in a separate draft, [yang-ted].

3.2. Layer 3 Unicast - IGP

The Layer 3 Unicast IGP topology model is defined by YANG module "l3-unicast-igp-topology". The model is depicted in the following diagram. Brackets enclose list keys, "rw" means configuration, "ro" operational state data, "?" designates optional nodes, "*" designates nodes that can have multiple instances. Parantheses enclose choice and case nodes. Notifications are not depicted. The prefix "nt:" refers to the YANG module for network topology.

```
module: l3-unicast-igp-topology
augment /nt:network-topology/nt:topology/nt:topology-types:
   +--rw l3-unicast-igp-topology!
augment /nt:network-topology/nt:topology:
   +--rw igp-topology-attributes
      +--rw name?   string
      +--rw flag*   flag-type
augment /nt:network-topology/nt:topology/nt:node:
   +--rw igp-node-attributes
      +--rw name?        inet:domain-name
      +--rw flag*        flag-type
      +--rw router-id*   inet:ip-address
      +--rw prefix* [prefix]
         +--rw prefix    inet:ip-prefix
         +--rw metric?   uint32
         +--rw flag*     flag-type
augment /nt:network-topology/nt:topology/nt:link:
   +--rw igp-link-attributes
      +--rw name?     string
      +--rw flag*     flag-type
      +--rw metric?   uint32
augment /nt:network-topology/nt:topology/nt:node/nt:termination-point:
   +--rw igp-termination-point-attributes
      +--rw (termination-point-type)?
         +--:(ip)
         |  +--rw ip-address*      inet:ip-address
         +--:(unnumbered)
            +--rw unnumbered-id?   uint32
```

The module augments the original network-topology module as follows:

- A new topology type is introduced, l3-unicast-igp-topology-type.

- Additional topology attributes are introduced, defined in a grouping, which augments the "topology" list of the network topology module. The attributes include an IGP name, as well as a set of flags (represented through a leaf-list). Each type of flag is represented by a separate identity. This allows to introduce additional flags in augmenting modules that are associated with specific IGP topologies, without needing to revise this module.

- Additional data objects for nodes are introduced by augmenting the "node" list of the network topology module. New objects include again a set of flags, as well as a list of prefixes. Each prefix in turn includes an ip prefix, a metric, and a prefix-specific set of flags.

- Links are augmented as well with a set of parameters, allowing to associate a link with an IGP name, another set of flags, and a link metric.

- Termination points are augmented with a choice of IP address or identifier.

In addition, the module defines a set of notifications to alert clients of any events concerning links, nodes, prefixes, and termination points. Each notification includes an indication of the type of event, the topology from which it originated, and the affected node, or link, or prefix, or termination point. In addition, as a convenience to applications, additional data of the affected node, or link, or termination point (respectively) is included. While this makes notifications larger in volume than they would need to be, it avoids the need for subsequent retrieval of context information, which also might have changed in the meantime.

3.3. OSPF Topology

OSPF is the next type of topology represented in the model. OSPF represents a particular type of Layer 3 Unicast IGP. Accordingly, this time the Layer 3 Unicast IGP topology model needs to be extended. The corresponding extensions are introduced in a separate YANG module "ospf-topology", whose structure is depicted in the following diagram. For the most part, this module augments "l3-unicast-igp-topology". Like before, brackets enclose list keys, "rw" means configuration, "ro" operational state data, "?" designates optional nodes, "*" designates nodes that can have multiple instances. Parantheses enclose choice and case nodes. Notifications respectively augmentations of notifications are not depicted. Likewise, used groupings from the TED module are not expanded but simply indicated by "...".

```
module: ospf-topology
augment /nt:network-topology/nt:topology/nt:topology-types/l3t:l3-unicast-igp-to
pology:
   +--rw ospf!
augment /nt:network-topology/nt:topology/l3t:igp-topology-attributes:
   +--rw ospf-topology-attributes
      +--rw area-id?   area-id
augment /nt:network-topology/nt:topology/nt:node/l3t:igp-node-attributes:
   +--rw ospf-node-attributes
      +--rw (router-type)?
      |  +--:(abr)
      |  |  +--rw abr?                 empty
      |  +--:(asbr)
      |  |  +--rw asbr?                empty
      |  +--:(internal)
      |  |  +--rw internal?            empty
      |  +--:(pseudonode)
      |     +--rw pseudonode?          empty
      +--rw dr-interface-id?     uint32
      +--rw multi-topology-id*   uint8
      +--rw capabilities?        bits
      +--rw ted
         +--...
augment /nt:network-topology/nt:topology/nt:link/l3t:igp-link-attributes:
   +--rw ospf-link-attributes
      +--rw multi-topology-id?   uint8
      +--rw ted
         +--...
augment /nt:network-topology/nt:topology/nt:node/l3t:igp-node-attributes/l3t:pre
fix:
   +--rw ospf-prefix-attributes
      +--rw forwarding-address?   inet:ipv4-address
```

The module augments "l3-unicast-igp-topology" as follows:

- A new topology type for an OSPF topology is introduced.

- Additional topology attributes are defined in a new grouping which augments igp-topology-attributes of the l3-unicast-igp-topology module. The attributes include an OSPF area-id identifying the OSPF area.

- Additional data objects for nodes are introduced by augmenting the igp-node-attributes of the l3-unicast-igp-topology module. New objects include router-type, dr-interface-id for pseudonodes, list of multi-topology-ids, ospf node capabilities, and traffic engineering attributes.

- Links are augmented with a multi-topology-id and traffic engineering link attributes.

- o Prefixes are augmented with OSPF specific forwarding address.

In addition, the module extends IGP node, link and prefix notifications with OSPF attributes.

3.4. IS-IS Topology

IS-IS is another type of Layer 3 Unicast IGP. Like OSPF topology, IS-IS topology is defined in a separate module, "isis-topology", which augments "l3-unicast-igp-topology". The structure is depicted in the following diagram. Like before, brackets enclose list keys, "rw" means configuration, "ro" operational state data, "?" designates optional nodes, "*" designates nodes that can have multiple instances. Parantheses enclose choice and case nodes. Notifications are not depicted. Likewise, used groupings from the TED module are not expanded but simply indicated by "...".

```
module: isis-topology
augment /nt:network-topology/nt:topology/nt:topology-types/l3t:l3-unicast-igp-to
pology:
   +--rw isis!
augment /nt:network-topology/nt:topology/l3t:igp-topology-attributes:
   +--rw isis-topology-attributes
      +--rw net?   iso-net-id
augment /nt:network-topology/nt:topology/nt:node/l3t:igp-node-attributes:
   +--rw isis-node-attributes
      +--rw iso
      |  +--rw iso-system-id?       iso-system-id
      |  +--rw iso-pseudonode-id?   iso-pseudonode-id
      +--rw net*                 iso-net-id
      +--rw multi-topology-id*   uint8
      +--rw (router-type)?
      |  +--:(level-2)
      |  |  +--rw level-2?            empty
      |  +--:(level-1)
      |  |  +--rw level-1?            empty
      |  +--:(level-1-2)
      |     +--rw level-1-2?          empty
      +--rw ted
         +--...
augment /nt:network-topology/nt:topology/nt:link/l3t:igp-link-attributes:
   +--rw isis-link-attributes
      +--rw multi-topology-id?   uint8
      +--rw ted
         +--...
```

The module augments the l3-unicast-igp-topology as follows:

- o A new topology type is introduced for isis.

- Additional topology attributes are introduced in a new grouping which augments "igp-topology-attributes" of the l3-unicast-igp-topology module. The attributes include an ISIS NET-id identifying the area.

- Additional data objects for nodes are introduced by augmenting "igp-node-attributes" of the l3-unicast-igp-topology module. New objects include router-type, iso-system-id to identify the router, a list of multi-topology-id, a list of NET ids, and traffic engineering attributes.

- Links are augmented with multi-topology-id and traffic engineering link attributes.

In addition, the module augments IGP nodes and links with ISIS attributes.

3.5. TED - Traffic Engineering Data

EDITOR’S NOTE: Module "ted" will be removed from subsequent revisions and be specified in a separate draft, [yang-ted]. Hence, this section will be removed from future revisions of this specification as well.

Traffic Engineering Data is required both by OSPF and IS-IS, which are defined in separate modules. Information shared by both is defined in another module, "ted". This module defines a set of groupings with auxiliary information required and shared by those other modules. This module details traffic-engineering node and link attributes:

- TED node attributes include te-router-id for IPv4 and IPv6, local IPv4 and IPv6 addresses and path computation client capabilities. The path computation client capabilities in turn include a bit vector for various path computation capabilities.

- TED link attributes comprise link color, max-link-bandwidth, max-resv-link-bandwidth, unreserved bandwidth and re-metric. They also include SRLG attributes which contains interface switching capabilities, a list of SRLG values, and a link protection type. The interface switching capabilities in turn contain a list element for each switching capability, defining encoding, max-lsp-bandwidth, and interface switching specific attributes.

4. Layer 3 Unicast IGP Topology YANG Module

```
   <CODE BEGINS>
   file "l3-unicast-igp-topology@2013-10-27.yang"
   module l3-unicast-igp-topology {
     yang-version 1;
     namespace "urn:TBD:params:xml:ns:yang:nt:l3-unicast-igp-topology";
     // replace with IANA namespace when assigned
     prefix "l3t";
     import network-topology {
       prefix "nt";
     }

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

     organization "TBD";
     contact "TBD";

     description
       "This module defines a model for the layer 3 IGP topology.";

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

     typedef igp-event-type {
       type enumeration {
         enum "add" {
           value 0;
           description
             "An IGP node or link or prefix or termination-point has
             been added";
         }
         enum "remove" {
           value 1;
           description
             "An IGP node or link or prefix or termination-point has
             been removed";
         }
         enum "update" {
           value 2;
           description
             "An IGP node or link or prefix or termination-point has
             been updated";
         }
       }
       description "IGP Event type for notifications";
     } // igp-event-type

     identity flag-identity {
       description "Base type for flags";
     }
     identity undefined-flag {
       base "flag-identity";
       description "Undefined flag";
     }

     typedef flag-type {
       type identityref {
         base "flag-identity";
       }
       description "Type for flags";
     }

     grouping topo-ref {
       description
         "Grouping for an absolute reference to a topology instance.";
       leaf topo-ref {
         type leafref {
           path "/nt:network-topology/nt:topology/nt:topology-id";
         }
         description
           "An absolute reference to a topology instance.";
       }
     }

     grouping link-ref {
       description
         "Grouping for an absolute reference to a link instance.";
       uses topo-ref;
       leaf link-ref {
         type leafref {
           path "/nt:network-topology/nt:topology"
             +"[nt:topology-id = current()/../topo-ref]"
             +"/nt:link/nt:link-id";
         }
         description
           "An absolute reference to a link instance.";
       }
     }

     grouping node-ref {
       description
```

```
         "Grouping for an absolute reference to a node instance.";
       uses topo-ref;
       leaf node-ref {
         type leafref {
           path "/nt:network-topology/nt:topology"
             +"[nt:topology-id = current()/../topo-ref]"
             +"/nt:node/nt:node-id";
         }
         description
           "An absolute reference to a node instance.";
       }
     }

     grouping tp-ref {
       description
         "Grouping for an absolute reference to a termination point.";
       uses node-ref;
       leaf tp-ref {
         type leafref {
           path "/nt:network-topology/nt:topology"
             +"[nt:topology-id = current()/../topo-ref]"
             +"/nt:node[nt:node-id = current()/../node-ref]"
             +"/nt:termination-point/nt:tp-id";
         }
         description
           "Grouping for an absolute reference to a termination point.";
       }
     }

     grouping igp-prefix-attributes {
       description
         "IGP prefix attributes";
       leaf prefix {
         type inet:ip-prefix;
         description "IP prefix value";
       }
       leaf metric {
         type uint32;
         description "Prefix metric";
       }
       leaf-list flag {
         type flag-type;
         description "Prefix flags";
       }
     }

     grouping l3-unicast-igp-topology-type {
       description "Identify the topology type to be L3 unicast.";
       container l3-unicast-igp-topology {
         presence "indicates L3 Unicast IGP Topology";
         description
           "The presence of the container node indicates L3 Unicast
           IGP Topology";
       }
     }

     grouping igp-topology-attributes {
       description "Topology scope attributes";
       container igp-topology-attributes {
         description "Containing topology attributes";
         leaf name {
           type string;
           description "Name of the topology";
         }
         leaf-list flag {
           type flag-type;
           description "Topology flags";
         }
       }
     }

     grouping igp-node-attributes {
       description "IGP node scope attributes";
       container igp-node-attributes {
         description "Containing node attributes";
         leaf name {
           type inet:domain-name;
           description "Node name";
         }
         leaf-list flag {
           type flag-type;
           description "Node operational flags";
         }
         leaf-list router-id {
           type inet:ip-address;
           description "Router-id for the node";
         }
         list prefix {
           key "prefix";
           description
             "A list of prefixes along with their attributes";
           uses igp-prefix-attributes;
         }
       }
     }
     grouping igp-link-attributes {
       description "IGP link scope attributes";
       container igp-link-attributes {
         description "Containing link attributes";
         leaf name {
           type string;
           description "Link Name";
         }
         leaf-list flag {
           type flag-type;
           description "Link flags";
         }
         leaf metric {
           type uint32 {
             range "0..16777215"  {
               description "
                           ";
               // OSPF/ISIS supports max 3 byte metric.
               // Ideally we would like this restriction to be
               // defined in the derived models, however,
               // we are not allowed to augment a "must" statement.
             }
           }
           description "Link Metric";
         }
       }
     } // grouping igp-link-attributes

     grouping igp-termination-point-attributes {
       description "IGP termination point scope attributes";
       container igp-termination-point-attributes {
         description "Containing termination point attributes";
         choice termination-point-type {
           description "Indicates the termination point type";
           case ip {
             leaf-list ip-address {
               type inet:ip-address;
               description "IPv4 or IPv6 address";
             }
           }
           case unnumbered {
             leaf unnumbered-id {
               type uint32;
               description "Unnumbered interface identifier";
             }
           }
         }
       }
     } // grouping igp-termination-point-attributes

     augment "/nt:network-topology/nt:topology/nt:topology-types" {
       description "Augment topology type on network-topology";
       uses l3-unicast-igp-topology-type;
     }

     augment "/nt:network-topology/nt:topology" {
       when "nt:topology-types/l3-unicast-igp-topology" {
         description "Augment only for L3 unicast IGP topology";
       }
       description "Augment topology configuration";
       uses igp-topology-attributes;
     }

     augment "/nt:network-topology/nt:topology/nt:node" {
       when "../nt:topology-types/l3-unicast-igp-topology" {
         description "Augment only for L3 unicast IGP topology";
       }
       description "Augment topology node configuration";
       uses igp-node-attributes;
     }

     augment "/nt:network-topology/nt:topology/nt:link" {
       when "../nt:topology-types/l3-unicast-igp-topology" {
         description "Augment only for L3 unicast IGP topology";
       }
       description "Augment topology link configuration";
       uses igp-link-attributes;
     }

     augment "/nt:network-topology/nt:topology/nt:node/"
            +"nt:termination-point" {
       when "../../nt:topology-types/l3-unicast-igp-topology" {
         description "Augment only for L3 unicast IGP topology";
       }
       description "Augment topology termination point configuration";
       uses igp-termination-point-attributes;
     }

     notification igp-node-event {
       description "Notification event for IGP node";
       leaf igp-event-type {
         type igp-event-type;
         description "Event type";
       }
       uses node-ref;
       uses l3-unicast-igp-topology-type;
```

```
        uses igp-node-attributes;
      }

      notification igp-link-event {
        description "Notification event for IGP link";
        leaf igp-event-type {
          type igp-event-type;
          description "Event type";
        }
        uses link-ref;
        uses l3-unicast-igp-topology-type;
        uses igp-link-attributes;
      }

      notification igp-prefix-event {
        description "Notification event for IGP prefix";
        leaf igp-event-type {
          type igp-event-type;
          description "Event type";
        }
        uses node-ref;
        uses l3-unicast-igp-topology-type;
        container prefix {
          description "Containing IPG prefix attributes";
          uses igp-prefix-attributes;
        }
      }

      notification termination-point-event {
        description "Notification event for IGP termination point";
        leaf igp-event-type {
          type igp-event-type;
          description "Event type";
        }
        uses tp-ref;
        uses l3-unicast-igp-topology-type;
        uses igp-termination-point-attributes;
      }
    }
    <CODE ENDS>
```

5. OSPF Topology YANG Module

```
    <CODE BEGINS>
    file "ospf-topology@2013-10-27.yang"
    module ospf-topology {
      yang-version 1;
      namespace "urn:TBD:params:xml:ns:yang:ospf-topology";
```

```
     // replace with IANA namespace when assigned
     prefix "ospf";

     import network-topology {
       prefix "nt";
     }

     import l3-unicast-igp-topology {
       prefix "l3t";
     }
     import ietf-inet-types {
       prefix "inet";
     }
     import ted {
       prefix "ted";
     }

     organization "TBD";
     contact "TBD";
     description "OSPF Topology model";

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

     typedef area-id {
       type uint32;
       description "OSPF Area ID";
     }

     grouping ospf-topology-type {
       description
         "Identifies the OSPF topology type.";
       container ospf {
         presence "indiates OSPF Topology";
         description
           "Its presence identifies the OSPF topology type.";
       }
     }

     augment "/nt:network-topology/nt:topology/nt:topology-types/"
       +"l3t:l3-unicast-igp-topology" {
       description
         "Defines the OSPF topology type.";
       uses ospf-topology-type;
     }
     augment "/nt:network-topology/nt:topology/"
       +"l3t:igp-topology-attributes" {
       when "../nt:topology-types/l3t:l3-unicast-igp-topology/ospf" {
         description "Augment only for OSPF topology";
       }
       description "Augment topology configuration";
       container ospf-topology-attributes {
         description "Containing topology attributes";
         leaf area-id {
           type area-id;
           description "OSPF area ID";
         }
       }
     }

     augment "/nt:network-topology/nt:topology/nt:node/"
       +"l3t:igp-node-attributes" {
       when "../../nt:topology-types/l3t:l3-unicast-igp-topology/ospf" {
         description "Augment only for OSPF topology";
       }
       description "Augment node configuration";
       uses ospf-node-attributes;
     }

     augment "/nt:network-topology/nt:topology/nt:link/"
       +"l3t:igp-link-attributes" {
       when "../../nt:topology-types/l3t:l3-unicast-igp-topology/ospf" {
         description "Augment only for OSPF topology";
       }
       description "Augment link configuration";
       uses ospf-link-attributes;
     }

     augment "/nt:network-topology/nt:topology/nt:node/"
       +"l3t:igp-node-attributes/l3t:prefix" {
       when "../../../nt:topology-types/l3t:l3-unicast-igp-topology/"
         +"ospf" {
         description "Augment only for OSPF topology";
       }
       description "Augment prefix";
       uses ospf-prefix-attributes;
     }

     grouping ospf-node-attributes {
       description "OSPF node scope attributes";
       container ospf-node-attributes {
         description "Containing node attributes";
         choice router-type {
           description "Indicates router type";
           case abr {
             leaf abr {
               type empty;
               description "The node is ABR";
             }
           }
           case asbr {
             leaf asbr {
               type empty;
               description "The node is ASBR";
             }
           }
           case internal {
             leaf internal {
               type empty;
               description "The node is internal";
             }
           }
           case pseudonode {
             leaf pseudonode {
               type empty;
               description "The node is pseudonode";
             }
           }
         }
         leaf dr-interface-id {
           when "../router-type/pseudonode" {
             description "Valid only for pseudonode";
           }
           type uint32;
           default "0";
           description "For pseudonodes, DR interface-id";
         }
         leaf-list multi-topology-id {
           type uint8 {
             range "0..127";
           }
           max-elements "128";
           description
             "List of Multi-Topology Identifier up-to 128 (0-127).
             RFC 4915";
         }
         leaf capabilities {
           type bits {
             bit graceful-restart-capable {
               position 0;
               description "Graceful restart capable";
```

```
            }
            bit graceful-restart-helper {
              position 1;
              description "Graceful restart helper";
            }
            bit stub-router-support {
              position 2;
              description "Stub router support";
            }
            bit traffic-engineering-support {
              position 3;
              description "Traffic engineering support";
            }
            bit point-to-point-over-lan {
              position 4;
              description "Support point to point over LAN";
            }
            bit experimental-te {
              position 5;
              description "Support experimental traffic engineering";
            }
          }
          description "OSPF capabilities as bit vector. RFC 4970";
        }
        container ted {
          description "Containing TE attributes";
          uses ted:ted-node-attributes;
        }
      } // ospf
    } // ospf-node-attributes

    grouping ospf-link-attributes {
      description "OSPF link scope attributes";
      container ospf-link-attributes {
        description "Containing OSPF link attributes";
        leaf multi-topology-id {
          type uint8 {
            range "0..127";
          }
          description "Muti topology ID";
        }
        container ted {
          description "Containing TE attributes";
          uses ted:ted-link-attributes;
        }
      }
    } // ospf-link-attributes
```

```
     grouping ospf-prefix-attributes {
       description "OSPF prefix attributes";
       container ospf-prefix-attributes {
         description "Containing prefix attributes";
         leaf forwarding-address {
           when "../../l3t:l3-unicast-igp-topology/l3t:ospf/"
             +"l3t:router-type/l3t:asbr" {
             description "Valid only for ABSR";
           }
           type inet:ipv4-address;
           description "Forwarding address for ABSR";
         }
       }
     }

     augment "/l3t:igp-node-event" {
       description "OSPF node event";
       uses ospf-topology-type;
       uses ospf:ospf-node-attributes;
     }

     augment "/l3t:igp-link-event" {
       description "OSPF link event";
       uses ospf-topology-type;
       uses ospf:ospf-link-attributes;
     }

     augment "/l3t:igp-prefix-event" {
       description "OSPF prefix event";
       uses ospf-topology-type;
       uses ospf:ospf-prefix-attributes;
     }
   }
   <CODE ENDS>
```

6. ISIS Topology YANG Module

```
<CODE BEGINS>
file "isis-topology@2013-10-21.yang"
module isis-topology {
    yang-version 1;
    namespace "urn:TBD:params:xml:ns:yang:network:isis-topology";
    // replace with IANA namespace when assigned
    prefix "isis";
    import network-topology {
        prefix nt;
    }
    import l3-unicast-igp-topology {
        prefix igp;
    }
    import ted {
        prefix ted;
    }

    organization "TBD";
    contact "TBD";
    description "ISIS Topology model";

    revision "2013-10-21" {
        description "Initial version";
    }
    typedef iso-system-id {
        description "ISO System ID. RFC 1237";
        type string {
            pattern '[0-9a-fA-F]{4}(\.[0-9a-fA-F]{4}){2}';
        }
    }

    typedef iso-pseudonode-id {
        description "ISO pseudonode id for broadcast network";
        type string {
            pattern '[0-9a-fA-F]{2}';
        }
    }
    typedef iso-net-id {
        description "ISO NET ID. RFC 1237";
        type string {
            pattern '[0-9a-fA-F]{2}((\.[0-9a-fA-F]{4}){6})';
        }
    }

    grouping isis-topology-type {
        container isis {
            presence "Indicates ISIS Topology";
        }
    }

    augment "/nt:network-topology/nt:topology/nt:topology-types/igp:l3-unicast-i
gp-topology" {
        uses isis-topology-type;
    }

    augment "/nt:network-topology/nt:topology/igp:igp-topology-attributes" {
        when "../nt:topology-types/l3t:l3-unicast-igp-topology/isis";
        container isis-topology-attributes {
            leaf net {
                type iso-net-id;
            }
        }
    }

    augment "/nt:network-topology/nt:topology/nt:node/igp:igp-node-attributes" {
        when "../../nt:topology-types/l3t:l3-unicast-igp-topology/isis";
        uses isis-node-attributes;
    }

    augment "/nt:network-topology/nt:topology/nt:link/igp:igp-link-attributes" {
        when "../../nt:topology-types/l3t:l3-unicast-igp-topology/isis";
        uses isis-link-attributes;
    }

    grouping isis-node-attributes {
        container isis-node-attributes {
            container iso {
                leaf iso-system-id {
                    type iso-system-id;
                }
                leaf iso-pseudonode-id {
                    default "0";
                    type iso-pseudonode-id;
                }
            }
            leaf-list net {
                max-elements 3;
                type iso-net-id;
            }
            leaf-list multi-topology-id {
                description "List of Multi Topology Identifier upto 128 (0-127).
 RFC 4915";
                max-elements "128";
                type uint8 {
                    range "0..127";
                }
            }
            choice router-type {
                case level-2 {
                    leaf level-2 {
                        type empty;
                    }
                }
                case level-1 {
                    leaf level-1 {
                        type empty;
                    }
                }
                case level-1-2 {
                    leaf level-1-2 {
                        type empty;
                    }
                }
            }
            container ted {
                uses ted:ted-node-attributes;
            }
        }
    }

    grouping isis-link-attributes {
        container isis-link-attributes {
            leaf multi-topology-id {
                type uint8 {
                    range "0..127";
                }
            }
            container ted {
                uses ted:ted-link-attributes;
            }
        }
    }

    augment "/igp:igp-node-event" {
        uses isis-topology-type;
        uses isis-node-attributes;
    }

    augment "/igp:igp-link-event" {
        uses isis-topology-type;
        uses isis-link-attributes;
    }
} // Module isis-topology

<CODE ENDS>
```

7. TED YANG Module

EDITOR’S NOTE: Module "ted" will be removed from subsequent revisions and be specified in a separate draft, [yang-ted].

```
<CODE BEGINS>
file "ted@2014-10-27.yang"
module ted {
  yang-version 1;
  namespace "urn:TBD:params:xml:ns:yang:network:ted";
  // replace with IANA namespace when assigned
```

```
     prefix ted;

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

     organization "TBD";
     contact
       "TBD";
     description
       "Helper module to hold trafic engineering attributes for
        any topology on which trafic engineering is enabled.";

     revision 2014-10-27 {
       description
         "Initial revision";
       reference
         "RFC 4220: Traffic Engineering Link Management Information
         Base.
         RFC 6825: Traffic Engineering Database Management Information
         Base in Support of MPLS-TE/GMPLS.
         RFC 4802: Generalized Multiprotocol Label Switching (GMPLS)
         Traffic Engineering Management Information Base";
     }

     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 "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 Capabilities of an interface.";
       reference
         "RFC 5307: IS-IS Extensions in Support of Generalized
          Multi-Protocol Label Switching (GMPLS).
          RFC 3812: Multiprotocol Label Switching (MPLS) Traffic
          Engineering (TE) Management Information Base (MIB).
          RFC 7074: Revised Definition of the GMPLS Switching Capability
          and Type Fields.
          RFC 7138: Traffic Engineering Extensions to OSPF
          for GMPLS Control of Evolving G.709 Optical Transport
          Networks.
          RFC 7139: GMPLS Signaling Extensions for Control of Evolving
          G.709 Optical Transport Networks.";
     }

     typedef encoding-type {
       type enumeration {
         enum "not-gmpls" {
           value 0;
           description
             "GMPLS is not in use";
         }
         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 or SONET";
         }
         enum "digital-wrapper" {
           value 7;
           description
             "Digital Wrapper";
         }
         enum "lambda" {
           value 8;
           description
             "Lambda";
         }
         enum "fiber" {
           value 9;
           description
             "Fiber";
         }
         enum "fiber-channel" {
           value 11;
           description
             "Fiber Channel";
         }
         enum "oduk" {
           value 12;
           description
             "G.709 OKUk (Digital Path)";
         }
         enum "optical-channel" {
           value 13;
           description
             "G.709 Optical Channel";
         }
       }
       description
         "Encoding type of an interface.";
       reference
         "RFC 3471: Generalized Multi-Protocol Label Switching (GMPLS)
          Signaling Functional Description.
          RFC 4328: Generalized Multi-Protocol Label Switching (GMPLS)
          Signaling Extensions for G.709 Optical Transport Networks
          Control.";
     }

     typedef pcc-capabilities {
       type bits {
         bit path-computation-with-gmpls-link-constraints {
           position 0;
           description "Link constraints";
         }
         bit bidirectional-path-computation {
           position 1;
           description "Bidirectional";
         }
         bit diverse-path-computation {
           position 2;
           description "Diverse";
         }
         bit load-balanced-path-computation {
           position 3;
           description "Load balanced";
         }
         bit synchronized-path-computation {
           position 4;
           description "Synchronized";
         }
         bit support-for-multiple-objective-functions {
           position 5;
           description "Multiple objective functions";
         }
         bit support-for-additive-path-constraints {
           position 6;
           description "Additive path constraints";
         }
         bit support-for-request-prioritization {
           position 7;
           description "Request prioritization";
         }
         bit support-for-multiple-requests-per-message {
           position 8;
           description "Multiple requests per message";
         }
       }
       description
         "Path Computation Capabilities.";
       reference
         "RFC 5088, draft-ietf-pce-disco-protoc-isis-07.txt
          OSPF/ISIS Protocol Extensions for Path Computation Element
          (PCE) Discovery.";
     }

     grouping ted-node-attributes {
       description
         "Identifier to uniquely identify a node in TED";
       reference
         "RFC 5305, RFC 6119: IPv6 Traffic Engineering in IS-IS/OSPF";
       leaf te-router-id-ipv4 {
         type inet:ipv4-address;
         description
           "Globally unique IPv4 Traffic Engineering Router ID.";
       }
       leaf te-router-id-ipv6 {
         type inet:ipv6-address;
         description
           "Globally unique IPv6 Traffic Engineering Router ID";
       }
       list ipv4-local-address {
         key "ipv4-prefix";
         description
           "List of IPv4 Local Address(OSPF). RFC 5786";

         leaf ipv4-prefix {
           type inet:ipv4-prefix;
           description
             "Local IPv4 address for the node";
         }
       }
       list ipv6-local-address {
         key "ipv6-prefix";
         description
           "List of IPv6 Local Address.";
         reference
           "RFC 5786: Advertising a Router’s Local Addresses
            in OSPF Traffic Engineering (TE) Extensions";
         leaf ipv6-prefix {
           type inet:ipv6-prefix;
           description
             "Local IPv6 address for the node";
         }
         leaf prefix-option {
           type uint8;
           description
             "IPv6 prefix option.";
         }
       }
       leaf pcc-capabilities {
         type pcc-capabilities;
         description
           "OSPF/ISIS PCC capabilities";
       }
     }

     grouping ted-link-attributes {
       description
         "TED Attributes associated with the link.";
       reference
         "RFC 3630, RFC 3784: IS-IS / OSPF Traffic Engineering (TE)";
       leaf link-index {
         type uint64;
         description
           "The link identifier.  If OSPF is used, this represents an
           ospfLsdbID.  If IS-IS is used, this represents an isisLSPID.
           If a locally configured link is used, this object represents
           an unique value, which is locally defined in a router.";
       }
       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 about the link";
       }
       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.";
       }
       leaf admin-status {
         type enumeration {
           enum up {
             value 1;
             description
               "Enabled.";
           }
           enum down {
             value 2;
             description
               "Disabled.";
           }
           enum testing {
             value 3;
             description
               "In some test mode.";
           }
           }
         description
           "The desired state of the link.";
         }
       leaf oper-status {
         type enumeration {
           enum up {
             value 1;
             description
               "Operational up";
           }
           enum down {
             value 2;
             description
               "Operational down.";
           }
           enum testing {
             value 3;
             description
               "In some test mode";
           }
           enum unknown {
             value 4;
             description
               "Status cannot be determined for some reason.";
           }
         }
         description
           "The current operational state of the link.";
       }
       leaf area-id {
         type binary {
           length 1..13;
         }
         description
           "This object indicates the area identifier of the IGP.
           If OSPF is used to advertise LSA, this represents an
           ospfArea. If IS-IS is used, this represents an area address.
           Otherwise, this is zero.";
       }
       leaf color {
         type uint32;
         description
           "Administrative group or color of the link";
       }
       leaf max-link-bandwidth {
         type decimal64 {
           fraction-digits 2;
         }
         description
           "Maximum bandwidth that can be see on this link in this
           direction. Units in bytes per second";
       }

       leaf max-resv-link-bandwidth {
         type decimal64 {
           fraction-digits 2;
         }
         description
           "Maximum amount of bandwidth that can be reserved in this
            direction in this link. Units in bytes per second";
       }
       list unreserved-bandwidth {
         key "priority";
         max-elements "8";
         description
           "Unreserved bandwidth for 0-7 priority levels. Units in
           bytes per second";
         leaf priority {
           type uint8 {
             range "0..7";
           }
           description "Priority";
         }
         leaf bandwidth {
           type decimal64 {
             fraction-digits 2;
           }
           description
             "Unreserved bandwidth for this level";
         }
       }
       leaf te-default-metric {
         type uint32;
         description
           "Traffic Engineering Metric";
       }
       leaf link-protection-type {
         type enumeration {
           enum "unprotected" {
             description "unprotected";
           }
           enum "extra-traffic" {
             description "Extra traffic";
           }
           enum "shared" {
             description "Shared";
           }
           enum "1-for-1" {
             description "One for one protection";
           }
           enum "1-plus-1" {
             description "One plus one protection";
           }
           enum "enhanced" {
             description "Enhanced protection";
           }
         }
         description
           "Link Protection Type desired for this link";
       }
       list interface-switching-capabilities {
         key "switching-capability";
         description
           "List of interface capabilities for this interface";
         leaf switching-capability {
           type ted:switching-capabilities;
           description
             "Switching Capability for this interface";
         }
         leaf encoding {
           type ted:encoding-type;
           description
             "Encoding supported by this interface";
         }
         list max-lsp-bandwidth {
           key "priority";
           max-elements "8";
           description
             "Maximum LSP Bandwidth at priorities 0-7";
           leaf priority {
             type uint8 {
               range "0..7";
             }
             description "Priority";
           }
           leaf bandwidth {
             type decimal64 {
               fraction-digits 2;
             }
             description
               "Max LSP Bandwidth for this level";
           }
         }
         container packet-switch-capable {
           when "../switching-capability = PSC-1  or "
             +"../switching-capability = PSC-2 or "
             +"../switching-capability = PSC-3 or "
             +"../switching-capability = PSC-4" {
             description "Valid only for PSC";
           }
           description
             "Interface has packet-switching capabilities";
           leaf minimum-lsp-bandwidth {
             type decimal64 {
               fraction-digits 2;
             }
             description
               "Minimum LSP Bandwidth. Units in bytes per second";
           }
           leaf interface-mtu {
             type uint16;
             description
               "Interface MTU";
           }
         }
         container time-division-multiplex-capable {
           when "../switching-capability = TDM" {
             description "Valid only for TDM";
           }
           description
             "Interface has time-division multiplex capabilities";
           leaf minimum-lsp-bandwidth {
             type decimal64 {
               fraction-digits 2;
             }
             description
               "Minimum LSP Bandwidth. Units in bytes per second";
           }
           leaf indication {
             type enumeration {
               enum "standard" {
                 description "Indicates support of standard SONET/SDH";
               }
               enum "arbitrary" {
                 description "Indicates support of arbitrary SONET/SDH";
               }
             }
             description
               "Indication whether the interface supports Standard or
                Arbitrary SONET/SDH";
           }
         }
       }
       container srlg {
         description
           "Shared Risk Link Group Attributes";
         uses srlg-attributes;
       }
     }

     grouping srlg-attributes {
       description
         "Shared Risk Link Group Attributes";
       reference
         "RFC 5307, RFC 4203: ISIS / OSPF Extensions in Support of
          Generalized Multi-Protocol Label Switching (GMPLS)";
       list srlg-values {
         key "srlg-value";
         description
           "List of Shared Risk Link Group this interface belongs to.";
         leaf srlg-value {
           type uint32;
           description
             "Shared Risk Link Group value";
         }
       }
     }
   }
   <CODE ENDS>
```

8. Security Considerations

The transport 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.

9. Contributors

The model presented in this paper was contributed to by more people than can be listed on the author list. Additional contributors include:

- Ken Gray, Juniper Networks

- Tom Nadeau, Brocade

- Aleksandr Zhdankin, Cisco

10. Acknowledgements

We wish to acknowledge the helpful contributions, comments, and suggestions that were received from Ladislav Lhotka, Andy Bierman, Carlos Pignataro, Joel Halpern, Juergen Schoenwaelder, and Alia Atlas.

11. References

11.1. Normative References

[RFC1195] Callon, R., "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments", RFC 1195, December 1990.

[RFC2178] Moy, J., "OSPF Version 2", RFC 2178, July 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.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, May 2014.

11.2. Informative References

[restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", I-D draft-ietf-netconf-restconf-03, October 2014.

[topology-dm]
Clemm, A., Medved, J., Tkacik, T., Varga, R., Bahadur, N., and H. Ananthakrishnan, "A YANG Data Model for Network Topologies", I-D draft-clemm-i2rs-yang-network-topo-01, October 2014.

[topology-use-cases]
Medved, J., Previdi, S., Lopez, V., and S. Amante, "Topology API Use Cases", I-D draft-amante-i2rs-topology-use-cases-01, October 2013.

[yang-json]
Lhotka, L., "JSON Encoding of Data Modeled with YANG", I-D draft-ietf-netmod-yang-json-01, October 2014.

[yang-ted]
Liu, X., Beeram, V., Clemm, A., Bryskin, I., and A. Guo, "A YANG module for traffic engineering database (TED)", I-D draft-liu-yang-ted-00, October 2014.

Authors’ Addresses

Alexander Clemm
Cisco

EMail: alex@cisco.com

Jan Medved
Cisco

EMail: jmedved@cisco.com

Robert Varga
Pantheon Technologies SRO

EMail: robert.varga@pantheon.sk

Tony Tkacik
Pantheon Technologies SRO

EMail: tony.tkacik@pantheon.sk

Xufeng Liu
Ericsson

EMail: xufeng.liu@ericsson.com

Igor Bryskin
Adva Optical

EMail: ibryskin@advaoptical.com

Aihua Guo
Adva Optical

EMail: aguo@advaoptical.com

Hariharan Ananthakrishnan
Packet Design

EMail: hari@packetdesign.com

Nitin Bahadur
Bracket Computing

EMail: nitin_bahadur@yahoo.com

Vishnu Beeram
Juniper Networks

EMail: vbeeram@juniper.net