Network Management Operations
Internet-Draft
Intended status: Informational
Expires: 20 September 2026

O. Havel
Huawei
N. Davis
Ciena
B. Claise
Huawei
O. G. D. Dios
Telefonica
T. Graf
Swisscom
19 March 2026

A YANG Data Model for SIMAP
draft-havel-nmop-simap-yang-02

Abstract

This document defines a YANG data model for Service & Infrastructure Maps (SIMAP). It extends the RFC8345 YANG modules to support all SIMAP requirements. This document will only focus on modelling proposal for each of the requirements not supported by RFC8345. Any related terminology, concepts, use cases and requirements are defined outside of this draft and this draft will only refer to them, analyze how to model and propose the implementation solutions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-nmop-simap-concept] defines Service & Infrastructure Maps (SIMAP) as a data model that provides a view of the operator’s networks and services, including how it is connected to other models (e.g., inventory), and data (e.g., observability data, operational knowledge). It specifically provides an approach to model multi-layered topology and an appropriate mechanism to navigate amongst layers and correlate between them. This includes layers from physical topology to service topology. This model is applicable to multiple domains (access, core, data center, etc.) and technologies (Optical, IP, etc.).

[I-D.ietf-nmop-simap-concept] defines the requirements for SIMAP, based on the Operator use cases. Some of the non-functional requirements should be supported by the protocols (NETCONF [RFC6241], RESTCONF [RFC8040] or other), by Network Configuration Access Control Model [RFC8431] and other current drafts in NETCONF WG. Some of the Operators’ requirements are already supported by the [RFC8345] ietf-network and ietf-network-topology YANG modules. Some of the Operators’ requirements identified the gaps in the ietf-network and ietf-network-topology YANG modules. This document proposes the solution how to implement these gaps.

2. Terminology

Terminology for SIMAP, SIMAP modelling, SIMAP data, topology, topology layer, multi-layered topology is defined in [RFC8345]

3. Why RFC8345 is a Good Approach for SIMAP Modelling

The primary reason for selecting RFC 8345 as the modeling framework is its simplicity and its ability to partially support the majority of core requirements.

RFC 8345 defines an abstract, generic, and foundational model for representing network and service topologies. It offers a straightforward mechanism for modeling networks and services as layered topologies, using four core entity types at each layer: network, node, link, and termination point. The model supports common relationships within the same layer, as well as underlay/overlay relationships across layers.

The following relationships are defined by RFC 8345:

* Relationships inside each layer are:

  - contains (cardinality 1:0..n)

    o a network contains nodes

    o a network contains links

    o a node contains termination points

  - source (cardinality: 1:0..1)

    o a link has one source termination point

  - destination (1:0..1)

    o a link has one destination termination point

* Relationships between layers, where underlay nodes, links and termination points must match underlay network

  - supporting for underlay (cardinality 0.._:0.._)

    o a network has supporting networks

    o a node has supporting nodes

    o a link has supporting links

    o a termination point has supporting termination points

Please note the following characteristics of the RFC 8345 topology model:

* Network type is used to identify the layer; sub-layers are not supported.

* Instantiation is flexible: a single network instance may represent one or multiple layers.

* The core entities — network, node, termination point, and link — are defined as abstract concepts and do not support role designation within the topology (e.g., root, leaf).

* The entities node, termination point and link are contained in a single network; they cannot be shared between the networks; they cannot connect different networks.

* Only unidirectional links are modeled, each with one source and one destination. Bidirectional connectivity must be represented using two separate unidirectional links.

* Layering relationships are expressed solely through the supporting construct, without additional semantics such as underlay, primary, backup, load-sharing, path, sequential, or parallel roles.

* The supporting relationship is restricted to same-type entities only: (network->network, node->node, link->link, termination point -> termination point)

* The model focuses exclusively on topology layering and does not support partitioning, such as sub-networks or sub-domains.

4. Requirements Categorization

The requirements from [I-D.ietf-nmop-simap-concept] were further analyzed and split into 3 categories:

* Section 4.1: Generic Requirements

  - these are generic SIMAP requirements for all YANG modules,

* Section 4.2: Requirements supported by RFC8345

  - these are SIMAP requirements already supported by [RFC8345] that do not require and extensions or modifications in RFC8345 for SIMAP modelling

* Section 4.3: Requirements not supported by RFC8345 (Gaps)

  - these are SIMAP requirements that are not supported by RFC8345 and are identified as RFC8345 gaps, they require extensions/ changes to ietf-network and ietf-network-topology YANG modules.

- the requirements in Section 4.3.2 must continue being supported when modelling the gaps.

This document version has a Section 4.4: Requirements for further analysis. These are the requirements that need further analyses in order to be categorized. This section will be removed from the future versions of the draft.

This document will focus on modelling the RFC8345 gaps and ensuring that the modelling changes do not break any requirements already supported by [RFC8345].

4.1. Generic requirements

The following requirements are generic interface requirements and do not impact SIMAP modelling:

* Design Requirements:
  - REQ-CONDITIONAL: Provide capability for conditional retrieval of parts of SIMAP. The NETCONF/RESTCONF and YANG support conditional retrieval. In the case that more advanced queries are needed, alternative query interface may be required.
* Architectural Requirements:
  - REQ-SCALES: The SIMAP API must be scalable.
  - REQ-PERFORMANCE: The SIMAP API must be performant.
  - REQ-SECURITY: The conventional NACM control access rules [RFC8341] should apply

4.2. Requirements supported by RFC8345

Based on our initial analysis, the following SIMAP requirements are already supported by RFC8345 and do not require any modelling extensions/modifications to ietf-network and ietf-network-topology:

* Operator Requirements:
  - REQ-BASIC-MODEL-SUPPORT: Basic model with network, node, link, and termination point entity types.
  - REQ-LAYERED-MODEL: Topology layers from physical layer up to service layer.

- REQ-VIEWPOINTS: Different viewpoints provide capability to have different views to different stakeholders.

- REQ-PASSIVE-TOPO: Topology includes passive topology.

- REQ-COMMON-API: Common SIMAP models and APIs, for multi domain.

- REQ-LIVE: Live network topology. TODO: Check if createTime / updatedTime / deletedTime is here or somewhere else.

- REQ-LAYER-NAVIGATE: Navigation inside the topology layer and between the topology layers

- REQ-DATA-PLANE-FLOW: Provider data plane (Flow) needs to be correlatable to underlay and customer data plane to overlay topology

- REQ-CONTROL-PLANE: Underlay control plane routing state needs to be correlatable to underlay L3 topology. Overlay control-plane routing state needs to be correlate-able to overlay L3 network topology.

* Design Requirements:

  - REQ-TOPO-ONLY: SIMAP should contain only topological information.

* Architectural Requirements:

  - REQ-DISCOVERY: A network controller must perform the initial and on-demand discovery of a network

  - REQ-SYNCH: The controller must perform the sync with the network.

  - REQ-USABILITY: The SIMAP API must be simple and easy to integrate with the client applications. This requirement is supported for the unidirectional and point2point networks.

We must keep these requirements in mind when proposing implementation solutions for gaps, as they are applicable to how we model the extensions / changes.

4.3. Requirements not supported by RFC8345 (Gaps)

Based on the initial analysis, the following SIMAP requirements are not fully supported by the RFC8345 and require extensions or modifications:

* Operator Requirements:

  - REQ-PROG-OPEN-MODEL: Open and programmable SIMAP.Gap: what-if and snapshots part. Analysis and solution presented in Section 7.4.1.

  - REQ-STD-API-BASED: Standard based SIMAP models and APIs, for multi-vendor support. Gap: link to external models. Analysis and solution presented in Section 7.5.1.

  - REQ-GRAPH-TRAVERSAL: Graph Traversal. Gap: links are entities, adding linkedTo relations would help. Analysis and solution presented in Section 7.3.1.

  - REQ-SNAPSHOT: Network snapshot topology. Analysis and solution presented in Section 7.4.2.

  - REQ-POTENTIAL: Potential new network topology. Analysis and solution presented in Section 7.4.3.

  - REQ-INTENDED: Intended topology. Analysis and solution presented in Section 7.4.4.

  - REQ-SEMANTIC: Network topology semantics. Gap: some semantic missing. Analysis and solution presented in Section 7.2.8.

  - REQ-EXTENSIBLE: Extensible via metadata. Analysis and solution presented in Section 7.2.2.

  - REQ-PLUGG: SIMAP must be pluggable. Analysis and solution presented in Section 7.5.2.

  - REQ-BIDIR: SIMAP must provide a mechanism to model bidirectional links. Gap: complex. Analysis and solution presented in Section 7.2.3.

  - REQ-MULTI-POINT: SIMAP must provide a mechanism to model multipoint links. Gap: complex. Analysis and solution presented in Section 7.2.4.

  - REQ-MULTI-DOMAIN: SIMAP must provide a mechanism to model links and nodes between networks. Analysis and solution presented in Section 7.2.5 and Section 7.2.6.

  - REQ-SUBNETWORK: SIMAP must provide a mechanism to model network decomposition into sub-networks. Analysis and solution presented in Section 7.2.7.

- REQ-SUPPORTING: SIMAP must provide a mechanism to model supporting relationships between different types of topological entities (e.g., a termination point is supported by the node). Gap: between different types. Analysis and solution presented in Section 7.2.9.

- REQ-STATUS: Links and nodes that are down must appear in the topology. Gap: optionally if status is in SIMAP, we need to model it.
Analysis and solution presented in Section 7.4.5.

* Design Requirements:

  - REQ-PROPERTIES: SIMAP entities should mainly contain properties used to identify topological entities at different layers, identify their roles, and topological relationships between them. Analysis and solution presented in Section 7.2.1.

  - REQ-RELATIONSHIPS: SIMAP should contain all topological relationships inside each layer or between the layers (underlay/overlay). Analysis and solution presented in Section 7.2.10.

  - REQ-TEMPO-HISTORY: Must support geo-spatial, temporal, and historical data. Analysis and solution presented in Section 7.4.6.

4.3.1. Features for RFC8345 Gaps

Based on the initial analysis, we identified that the requirements can be grouped into the following features * SIMAP Core. These are the core requirements for the SIMAP, applicable to the current topology snapshot and needed to support the live SIMAP model * REQ-PROPERTIES * REQ-EXTENSIBLE * REQ-BIDIR * REQ-MULTI-POINT * REQ-MULTI-DOMAIN * REQ-SUBNETWORK * REQ-SEMANTIC * REQ-SUPPORTING * REQ-RELATIONSHIPS

* SIMAP Navigation. These are the requirements for navigation of SIMAP * REQ-GRAPH-TRAVERSAL

* SIMAP Lifecycle. These are the requirements for modelling the lifecycle of the topology, including intended topology, potential topology, historical snapshots and relations between them. * REQ-PROG-OPEN-MODEL * REQ-SNAPSHOT * REQ-POTENTIAL * REQ-INTENDED * REQ-STATUS * REQ-TEMPO-HISTO

* SIMAP External Link. These are the requirements for linking the external models and data * REQ-STD-API-BASED * REQ-PLUGG

Please note that the SIMAP core feature may be further split during the further analysis and reviews of the SIMAP module. Analysis and solution presented in Section 7.1.

4.3.2. Requirements to keep in mind when modelling gaps

Any extensions/modifications must keep the original RFC8345 approach as simple as possible and fully generic and technology and layer agnostic. The following requirements are already supported in RFC8345, but we must keep them in mind when proposing implementation solutions for gaps, as they are applicable to how we model the extensions / changes:

* Operator Requirements:
    - REQ-BASIC-MODEL-SUPPORT: Basic model with network, node, link, and termination point entity types.
    - REQ-LAYERED-MODEL: Topology layers from physical layer up to service layer.
    - REQ-COMMON-API: Common SIMAP models and APIs, for multi domain.
    - REQ-LIVE: Live network topology. TODO: Check if createTime / updatedTime / deletedTime is here or somewhere else.
    - REQ-LAYER-NAVIGATE: Navigation inside the topology layer and between the topology layers
* Design Requirements:
    - REQ-TOPO-ONLY: SIMAP should contain only topological information.
* Architectural Requirements:
    - REQ-USABILITY: The SIMAP API must be simple and easy to integrate with the client applications. This requirement is supported for the unidirectional and point2point networks.

4.4. Requirements for further analysis

The following requirements need to be analyzed further to determine if they can be supported by RFC8345:

* Operator Requirements:

- REQ-TOPOLOGY-ABSTRACTION: Navigation across the abstraction levels inside a single network layer

The following requirements have to be analyzed further to capture all generic networking semantics missing from RFC8345 and from additional requirements defined as gaps.

* Operator Requirements:

  - REQ-SEMANTIC: Network topology semantics.

This section will be removed from the future versions of the draft.

5. Modelling approach in this draft version

There are multiple options where the modelling changes not supported by RFC8345 can be done. The following are candidate approaches:

* via new SIMAP YANG module, augmenting RFC8345 YANG modules.

* via RFC8345 bis YANG modules.

* via new SIMAP model that does not augment RFC8345 YANG modules (either new YANG module or some other way).

* different approach for different groups of requirements.

This draft introduces the new ietf-simap-topology YANG module that defines augmentation for ietf-network and ietf-network-topology nodes to support all SIMAP requirements. As this is the simplest way to analyze the gaps and review the modelling, the current version of this draft uses this approach. This does not mean that this is the proposed approach, it will be used for review of the proposed modelling changes until the final approach has been agreed in the NMOP. The final approach can be decided at the later stage, after the modelling is reviewed and proposed. The modelling can then be moved to the agreed module.

By proposing YANG models to address SIMAP requirements, we do not suggest that this is the only approach to modeling SIMAP. SIMAP can also be represented using ontological frameworks; however, that alternative is beyond the scope of this document.

6. API Scope

It has been identified from the use cases that there is need for the following:

* read access to the live topology (REQ-LIVE)

* read access to historical topology (REQ-SNAPSHOT)

* read and write access for potential topology (REQ-POTENTIAL)

* read and write access for intended topology (REQ-INTENDED)

Traditionally, separate API endpoints are used for accessing the live network and potential network views. If we follow this approach, the following consequences apply:

* The same topology model can be reused across live, potential, and intended network representations.

* Distinct API endpoints are maintained for each of the three categories.

* Access control (e.g., read-only for live and read/write for potential networks) can be managed externally via NACM access control rules as defined in [RFC8341], eliminating the need to use config true/false within the model itself.

* The responsibility for tracking relationships between live, potential, intended networks, and historical snapshots is delegated to applications. These applications must orchestrate across multiple APIs to compute differences between snapshots.

To support more advanced solutions and enable SIMAP to be positioned across various interfaces, as discussed in (Chapter 5 of [I-D.ietf-nmop-simap-concept], the current draft proposes the following enhancements:

* Define a default behavior that allows separation between live, potential, and intended topologies without embedding this semantic distinction in the model itself. Instead, semantics are determined by the selected API endpoint, and the model remains agnostic to the category of topology.

* Optionally enable a unified API that introduces model-level semantics to distinguish between live, potential, and intended topologies. This unified model also captures relationships among live, potential, historical, and intended instances. Such an approach supports advanced use cases that require the model to explicitly represent these semantics.

7. Solution Proposal for the RFC8345 Gaps

7.1. SIMAP Features

We are introducing a set of SIMAP features that will be used to group requirements and optionally include them in the SIMAP module if the feature is implemented. Definitions will be tagged with the feature name and are only valid on a SIMAP server that supports that feature.

The following is the initial set of SIMAP features:

```
module ietf-simap-topology {
    /* … other statements … */

        feature simap-core-topology {
                description
                    "This feature indicates that the SIMAP server supports those
             SIMAP requirements not supported by RFC8345, but related
             to core live topology";
        }

        feature simap-navigation {
                description
                    "This feature indicates that the SIMAP server supports
             those SIMAP requirements not supported by RFC8345, but
             related to connecting SIMAP navigation";
        }

        feature simap-lifecycle {
                description
                    "This feature indicates that the SIMAP server supports
             those SIMAP requirements not supported by RFC8345, but
             related to the overall SIMAP lifecycle, including intent,
             potential (for what-if) and historical snapshots";
        }

        feature simap-external {
                description
                    "This feature indicates that the SIMAP server supports
             those SIMAP requirements not supported by RFC8345, but
             related to connecting SIMAP to external models and data";
        }


    /* … other statements … */
}
```

Figure 1: The SIMAP Features

7.2. Feature simap-core-topology

The solution is proposed for the following requirements: * REQ-PROPERTIES
* REQ-EXTENSIBLE * REQ-BIDIR * REQ-MULTI-POINT * REQ-MULTI-DOMAIN * REQ-SUBNETWORK * REQ-SEMANTIC * REQ-SUPPORTING * REQ-RELATIONSHIPS

The changes to the RFC8345 model for these requirements would be specified inside: if-feature simap-core-topology.

7.2.1. REQ-PROPERTIES: Properties significant for topology

7.2.1.1. Analysis

This design requirement states that SIMAP entities should contain properties used to identify topological entities at different layers, identify their roles, and topological relationships between them. Multiple operator requirements are already covering some specifics, like REQ-BIDIR, REQ-MULTI-POINT and we will analyze and propose implementation for any missing semantic as part of REQ-SEMANTIC.

Nevertheless, sometimes the topological information is in the textual form and cannot be done via the model and is communicated via name, description, labels. We can also see that many current augmentations of RFC8345 add some common properties that are generic and needed at all layers and technologies (e.g. name).

Therefore, we propose to add name, labels and decription into the core model for all SIMAP entities but keep them optional for backward compatibility.

7.2.1.2. Implementation Proposal

```
  /*
   * Common SIMAP groupings for optional RFC8345 extensions
   * Addressing requirements REQ-PROPERTIES as name, label and
   * description may be important properties that
   * clarify the topological roles for different layers and technologies
   */

  grouping simap-common {
    description "A reusable set of optional extensions for network,
                 node, termination point and link";
    leaf name {
          if-feature simap-core-topology;
      type string;
      description
        "The user friendly name, if required.
        It is optional, for backward compatibility";
    }
    leaf-list label {
          if-feature simap-core-topology;
      type string;
      description
        "Used for optionally adding any labels to the instances,
         if required";
    }
    leaf description {
          if-feature simap-core-topology;
      type string;
      description
        "Used for optionally adding any description to the instances,
        if required";
    }

        TODO: Check if createTime / updatedTime / deletedTime is here or
somewhere else.

    :
    :


  }
```

Figure 2: The proposal for bidirectional links

7.2.2. REQ-EXTENSIBLE: Extensible via metadata

7.2.2.1. Analysis

SIMAP must be extensible with metadata. This metadata can be added via augmentation, but the model could also allow the approach of adding the metadata via the core topology model. This approach can be used for adding any semantics not explicitely implemented in the model.

7.2.2.2. Implementation Proposal

Add the following to network, node, termination-point and link, and to supporting relations:

```
  grouping simap-common {
    description "A reusable set of optional extensions for network,
                 node, termination point and link";

    :
    :
    anydata extension {
          if-feature simap-core-topology;
      config false;
      description
        "The extension point for any other meta info or data or any
        unknown extensions. Proposed solution for SIMAP requirement
        REQ-EXTENSIBLE. Any additional info that is topologically
        significant can also be added this way for requirement
        REQ-SEMANTIC or REQ-PROPERTIES.";
    }
  }
```

Figure 3: The proposal for extensions for meta data

7.2.3. REQ-BIDIR: Bidirectional Links

7.2.3.1. Analysis

One of the core characteristics of any network topology is the link directionality. While data flows are unidirectional, the bidirectional links are also common in networking. Examples are Ethernet cables, bidirectional SONET rings, socket connection to the server. We also encountered requirements for simplified service layer topology, where we want to model link as bidirectional to be supported by unidirectional links at the lower layer.

* RFC8345 supports only unidirectional links, it was done intentionally to keep the model as simple as possible

* RFC8345 suggests to model bidirectional links via multiple instances of unidirectional links

* while keeping the model simpler in RFC8345, the APIs and data become more complex and error prone. There is increase of number of instances / data transferred over API, stored in the DB, or managed/monitored and some errors may occur during creation of the entities

* while keeping the model simpler in RFC8345, we lack the semantics for the bidirectional links and capability to expose and create them in the unified manner

7.2.3.2. Implementation Proposal

```
identity link-direction {
  description "Base identity for the directionality of the link";
}

identity link-direction-uni {
  base "st:link-direction";
  description "Unidirectional link";
}

identity link-direction-bi {
  base "st:link-direction";
  description "Bidirectional link";
}

identity tp-direction {
  description "Base identity for the directionality of the tp";
}

identity tp-direction-symmetric {
  base "st:tp-direction";
  description "TP in the bidirectional link";
}

identity tp-direction-source {
  base "st:tp-direction";
  description "TP is the source tp in the link";
}

identity tp-direction-destination {
  base "st:tp-direction";
  description "TP is the destination tp in the link";
}
```

```
   augment "/nw:networks/nw:network/nt:link" {

     :
     :
     leaf link-direction {
           if-feature simap-core-topology;
       type identityref {
         base st:link-direction;
       }
       description
         "The direction of the link: unidirectional (link-direction-uni)
         or bidirectional (link-direction-bi). It can also be any other
         custom identity defined with base link-direction. It is
         optional, so the model supports the solution without the link
         direction information, either if not known or for backward
         compatible case.";
     }

     list tp {
           if-feature simap-core-topology;
       key "network-ref node-ref tp-ref";
       description "List of termination points in the link";
       uses nt:tp-ref;
         :
         :
       leaf tp-direction {
         type identityref {
           base st:tp-direction;
         }
         description
           "The direction of the point in the link";
       }
   }
```

Figure 4: The proposal for bidirectional links

7.2.4. REQ-MULTI-POINT: Multipoint Links

7.2.4.1. Analysis

One of the core characteristics of any network topology is its type and link cardinality. Any topology model should be able to model any topology type in a simple and explicit way, including point to multipoint, bus, ring, star, tree, mesh, hybrid and daisy chain. Any topology model should also be able to model any link cardinality in a simple and explicit way, including point to point, point to multipoint, multipoint to multipoint or hybrid.

* RFC8345 defines all links as point to point and unidirectional, it does not support multi-point links (hub and spoke, full mesh, hybrid). It was done intentionally to keep the model as simple as possible.

* RFC8345 suggests to model the multi-point networks via pseudo nodes.

* while keeping the model simpler in RFC8345, the APIs and data become more complex and error prone. There is increase of number of instances / data transferred over API, stored in the DB, or managed/monitored.

* while keeping the model simpler in RFC8345, we lack the semantics for multi-point links and capability to expose and create them in the unified manner

Clarification on hybrid links: The multi-point link is an abstraction of underlying structure detail. The underlying structure is currently expressed in terms of a simple identity which essentially references
a pattern of underlying flow where that pattern describes the effect (another abstraction) of that flow between each point of the link. There are currently a few well understood patterns, but the list will expand. The list of patterns, not surprisingly, will be the same as for service, for connectivity and for flow (as the link arises from them). The approach of introdutcion of hybrid links allows for a solution to express a complex arrangement in abstraction without needing, for each instance, to lay out the detail. It also allows the client application to “understand” the effect without having to probe that detail.

7.2.4.2. Implementation Proposal

```
identity link-type {
    description
      "Base identity for the internal structure of the link";
}

identity link-type-p2p {
    base "st:link-type";
    description "Point to point link";
}

identity link-type-p2mp {
    base "st:link-type";
    description "Point to multi-point link";
}
```

```
  identity link-type-mp2mp {
      base "st:link-type";
      description "Multi-point to multi-point link";
  }

  identity link-type-hybrid {
      base "st:link-type";
      description
        "Hybrid links, combination of the other three.
        The multi-point link is an abstraction of underlying structure
        detail. The underlying structure is currently expressed in
        terms of a simple identity which essentially references
        a pattern of underlying flow where that pattern describes the
        effect (another abstraction) of that flow between each point
        of the link. There are currently a few well understood
        patterns, but the list will expand. The list of patterns, not
        surprisingly, will be the same as for  service, for
        connectivity and for flow (as the link arises from them). The
        approach of introduction of hybrid links allows for a
        solution to express a complex arrangement in abstraction without
        needing, for each instance, to lay out the detail. It also
        allows the client application to “understand” the effect
        without having to probe that detail.";
  }

  augment "/nw:networks/nw:network/nt:link" {

    :
    :
    leaf link-type {
          if-feature simap-core-topology;
      type identityref {
        base st:link-type;
      }
      description
        "The reference to the specification for the internal structure
        of the link. It can be point to point (link-type-p2p), point to
        multipoint (link-type-p2mp) or multipoint to multipoint
        (link-type-mp2mp). It can also be any other custom identity
        defined with base link-type. It is optional, so the model
        supports the solution without the link type information,
        either if not known or for backward compatible case.";
    }
    list tp {
          if-feature simap-core-topology;
      key "network-ref node-ref tp-ref";
      description "List of termination points in the link";
      uses nt:tp-ref;
```

```
      leaf tp-role {
        type identityref {
          base st:tp-role;
        }
        description
          "The role of the termination point in the link defined in the
           link-type spec.";
        }
    :
    :
    }

  }
```

Figure 5: The proposal for multi-point links

On the stand-alone point to point unidirectional case (the case supported by RFC8345 currently), we recommend that, in a new deployment, this is represented simultaneously in terms of the existing approach and the new approach to bridge between old and new solutions. We can explore this evolution in more detail as it is very important that we get this right. In the longer term, it would be ideal if all new solutions used the proposed multi-point approach for point to point (where point to point is simply a represented with a list of two points). Clearly, as always, there will be a long tail on the legacy in the network, so some adaptation will be required to deal with older deployments, but this is usual and expected. It should not prevent advancement.

Open Issue: Comment received to address: To improve clarity, would it make sense to define rules? For example:

* point-to-point: 1-to-1

* point-to-multipoint: 1-to-2+

* multipoint-to-multipoint: 2+-to-2+

* And possibly multipoint-to-point: 2+-to-1 (if such cases are to be supported)

7.2.5. REQ-MULTI-DOMAIN: Multi-domain Links

7.2.5.1. Analysis

Link between multiple networks, sub-networks, or domains is the common concept in network topology. SIMAP must provide a mechanism to model links between networks.

* RFC8345 defines all links as belonging to one network instance and having the source and destination as node and termination point only, not allowing to link to termination point of another network.

* This does not allow for links between networks in the case of multi-domains or partitioning.

* The only way would be to model each domain as node and have links between them

7.2.5.2. Implementation Proposal

Allows the link to terminate on the termination point that is on another network.

```
augment "/nw:networks/nw:network/nt:link" {

  list tp {
        if-feature simap-core-topology;
    key "network-ref node-ref tp-ref";

  }
}
```

Figure 6: The proposal for multi-domain links

This way we can model the links in the following ways: 1. link belongs to one network (e.g. IS-IS Area A) but pointing to remote point of another network (e.g. IS-IS Area B) 2. link belongs to the parent domain (e.g. IS-IS AS Domain), but points to termination points of children domains (e.g.
IS-IS Area A and IS-IS Area B)

7.2.6. REQ-MULTI-DOMAIN: Multi-domain Nodes

7.2.6.1. Analysis

Nodes shared between multiple domains is the common concept in network topology (e.g. Area Border Routers that connect different areas in OSPF). SIMAP must provide a mechanism to model nodes shared between networks.

* RFC8345 defines all nodes as belonging to one network instance, not allowing to have a node inside the 2 networks.

* This does not allow for sharing a node between networks in the case of
  multi-domains or partitioning.

* Current approach is to either: * model a node with 2 instances, one per domain, which in the case of OSPF would not reflect the topology * have the same instance in 2 networks, only if node-id is unique on all nodes in all networks

7.2.6.2. Implementation Proposal

For further study, implement different candidates via hackathon

7.2.7. REQ-SUBNETWORK: Subnetworks and partitioning

7.2.7.1. Analysis

RFC8345 does not model networks being part of other networks, therefore cannot model subnetworks and network partitioning. We encountered this problem with modelling IS-IS and OSPF domains and areas. The goal is to model AS/domain with multiple areas so that the SIMAP model contains information about how the AS is first split into different IGP domains and how each IGP domain is split into different areas. This is a common problem for both IS-IS and OSPF.

7.2.7.2. Implementation Proposal

The following are the identified implementation candidates:

* Candidate 1 - network->subnetwork direction (currently proposed in the YANG module):

```
augment "/nw:networks/nw:network" {
  list subnetwork {
        if-feature simap-core-topology;
      key "network-ref";
         description
           "A subnetwork of the network, supports partitioning";
         leaf network-ref {
           type leafref {
             path "/nw:networks/nw:network/nw:network-id";
           require-instance false;
           }
           description
             "References the subnetworks";
         }
  }
}
```

Figure 7

* Candidate 2 - subnetwork->network direction:

```
augment "/nw:networks/nw:network" {
  leaf parent-ref {
        if-feature simap-core-topology;
    type leafref {
      path "/nw:networks/nw:network/nw:network-id";
      require-instance false;
    }
    description "References the parent network from subnetworks";
  }
}
```

Figure 8

7.2.8. REQ-SEMANTIC: Network Topology Semantic

7.2.8.1. Analysis

Many RFC8345 augmentations add the additional topological semantics, with same concepts modelled in a different way in different augmentations. We already mentioned that some semantic is missing from the RFC8345 topology model, like bidirectional and multi-point. The following is also missing from the model:

* Relationship Properties. The supporting relationship could have additional attributes that give more information about the supporting relationship. That way we could use it for aggregation, underlay with primary/backup, load balancing, hop, sequence, etc.

* Termination point roles. We are missing semantics for the common topology roles: uni, i-nni,e-nni, primary, backup, hub, spoke,

* Node roles. We are missing semantics for the common node roles: access, core, metro

* Link roles. We are missing semantics for the common link roles: uni, i-nni, e-nni

* Layers / Sublayers. We need further analysis to determine in network types are sufficient to support all scenarios for layers/ sublayers. Alternatevely, some sub-layer info needs to be added.

* Tunnels and paths. We can model tunnels and paths via RFC8345 but we loose some semantics that is in RFC8795.

7.2.8.2. Implementation Proposal

For further study, implement different candidates via hackathon

7.2.9. REQ-SUPPORTING: Extension to supporting

7.2.9.1. Analysis

RFC8345 defines supporting relationships only between the same type of entities. Networks can only be supported by networks, nodes can only be supported by nodes, termination points can only be supported by terminations points and links can only be supported by links.

During the hackathons, we had a scenario where at one layer of topology we had a link with termination points where the termination points are logical without the underlay termination point, but the only underlay connection we were able to define was to the underlay nodes. The same happened with nodes and networks.

Therefore, we encountered the need to have termination points supported by nodes and nodes supported by networks. The [RFC8795] also adds additional underlay relationship between node and topology and link and topology, but via a new underlay topology and not via the core supporting relationship.

7.2.9.2. Implementation Proposal

For further study, implement different candidates via hackathon

7.2.10. REQ-RELATIONSHIPS: Relationships significant for topology

7.2.10.1. Analysis

The following relations are already supported by RFC8345:

* containment * networks contain nodes * nodes contain termination points

7.2.10.2. Implementation Proposal

For further study, implement different candidates via hackathon

7.3. Feature simap-navigation

The solution is proposed for the following requirements: * REQ-GRAPH-TRAVERSAL

The changes to the RFC8345 model for these requirements would be specified inside: if-feature simap-navigation

7.3.1. REQ-GRAPH-TRAVERSAL: Graph Traversal

7.3.1.1. Analysis

RFC8345 states that one of the reasons for modelling only unidirectional and point to point links is to allows the model to be very easily subjected to applications that make use of graph algorithms.

If we add support for bidirectional and multipoint links, this will make the graph traversal more difficult according to RFC8345. But based on the discussions with some Operators, having links as nodes and not direct relations is not graph traversal friendly for paths in any case and having the capability to have the direct link relations between the termination points or nodes is needed.

The declarative graph queries will require the graph query language like Cypher or SPARQL. The current IETF modelling and protocols can only provide the limited graph navigation.

7.3.1.2. Implementation Proposal

Add read only linked-termination-point and linked-node for optimizing path graph traversal at single layer, keep them optional for backward compatibility.

```
     augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
       description
         "Adding optional read only link relations between connected
  termination points for optimized graph traversal for paths";

       list linked-termination-point {
             if-feature simap-navigation;
         config false;
         key "network-ref node-ref tp-ref";
         description
               "This list identifies any linked termination points, added
                optionally for read only for optimized path graph
                traversal";

         uses nt:tp-ref;
     }

     augment "/nw:networks/nw:network/nw:node" {
       description
         "Adding optional read only link relations between connected
  nodes for optimized graph traversal for paths";

       list linked-node {
             if-feature simap-navigation;
         config false;
         key "network-ref node-ref";
         description
               "This list identifies any linked nodes, added
                optionally for read only for optimized path graph
                traversal";

         uses nw:node-ref;
     }
```

Figure 9: The proposal for optimized path graph traversal

7.4. Feature simap-lifecycle

The solution is proposed for the following requirements: * REQ-PROG-OPEN-MODEL * REQ-SNAPSHOT * REQ-POTENTIAL * REQ-INTENDED * REQ-STATUS * REQ-TEMPO-HISTO

The changes to the RFC8345 model for these requirements would be specified inside: if-feature simap-lifecycle

7.4.1. REQ-PROG-OPEN-MODEL: Open and Programmable

7.4.1.1. Analysis

RFC8345 already supports the open and programmable API, but this requirements also mentions the what-if scenarios and different snapshots that may have some modelling implications (e.g. the need to switch between different snapshots, relation to the real network that the snapshot is for).

7.4.1.2. Implementation Proposal

The implementation for the model changes for this requirement will be covered by the implementation proposals for:

* REQ-POTENTIAL in Section 7.4.3 for write access to the potential new topology.

* REQ-INTENDED in Section 7.4.4 for write access for intended topologies.

7.4.2. REQ-SNAPSHOT: Different snapshots

7.4.2.1. Analysis

SIMAP must enable retrieval of multi-layered topology of different historical snapshots, where a snapshot is the view of the network at any given point in time. This requires the implementation proposal to take into account the following:

* how to retrieve historical snapshot for the network using time stamp (recorder time and/or validity time)

* how to retrieve historical snapshots for the current topology

* how to model and retrieve historical topology: either full topology or only changes from another snapshot

* how to retrieve the external models via links in the historical context

* is this SIMAP requirement or generic requirement for any YANG retrieval via NETCONF/RESTCONF. The [RFC8342] defines different datastores: configuration datastores (startup, candidate, running, intended) and operational datastore. It does not support the historical datastores.

7.4.2.2. Implementation Proposal

The following are the identified implementation candidates:

* Candidate 1 - Generic Solution:

  - this is generic requirement, not just for SIMAP, therefore implementation should not be implemented by SIMAP but by the NETCONF/RESTCONF (in NMDA?) for historical system state + conf. For conf changes via API there is history of transactions, but nothing we are aware of for historical of any data model.

* Candidate 2- SIMAP Specific Solution via historical files (currently proposed in the YANG module):

  - implemented via some historical files retrieval of the historical files for specified snapshot-id, network-id and timestamp, and the file itself can have the same structure as a member of the list network:

```
 container networks-history {
   if-feature simap-lifecycle;
   config false;
   list network-history {
     key "live-network-ref timestamp";
     description
       "Historical network. Snapshot is generated for each historical
        instance, network-ref is the reference for the live network";
     leaf live-network-ref {
        type leafref {
          path "/nw:networks/nw:network/nw:network-id";
     }
     leaf timestamp {
       type yang:date-and-time;
     }
     leaf file-id {
       type inet:uri;
     }
   }
 }
```

Figure 10: The candidate 2 for historical snapshots

* Candidate 3 - SIMAP Specific Solution via separate containers"

  - implemented via separate container, for read only. Instead of referencing the file, the whole yang structure for the network from the list-network is duplicated, except that all supporting references must also point to snapshot id as well. This implementation is not proposed in the yang file in this draft version.

```
container networks-history {
  if-feature simap-lifecycle;
  config false;
  list network-history-snapshot {
    key "network-ref snapshot-id"
    uses nt:network-ref;
    leaf snapshot-id {
      type inet:uri;
    }
    leaf timestamp {
      type yang:date-and-time;
    }
    container network-types {
    }
    list supporting-network {
        key "network-ref snapshot-id"
        leaf network-ref {}
        leaf snapshot-id {}
    }
    list node {
        leaf node-id {..}
        list supporting-node {
          key "network-ref snapshot-id node-ref";
          ...
        }
        list termination-point {
          leaf tp-id {}
          list supporting-termination-point {
            key "network-ref snapshot-id node-ref tp-ref";
            ...
          }
        }
    }
    list link {
        leaf link-id {..}
        list supporting-link {
          key "network-ref snapshot-id link-ref";
          ...
        }
    }
}
```

Figure 11: The candidate 3 for historical snapshots

The candidate 2 with file reference is currently implemented in the YANG module, but our preference would be to use a generic mechanism for historical retrieval, as this is not SIMAP only requirement but also applicable to inventory and other modules.

7.4.3. REQ-POTENTIAL: Potential new network topology

7.4.3.1. Analysis

SIMAP must enable both retrieval and write access to the potential new network. A potential new network is the view at a given point with modifications from the snapshot. This view may contain either the full topology or just differences from the snapshot. Running a "what-if" analysis requires the ability to take snapshots and to switch easily between them.

This requires the implementation proposal to take into account the following:

* SIMAP write is used for either what-if or intended scenarios, but we have to ensure that we can create multiple snapshots for what-if scenario

* how to retrieve different what-if network instances

* how to connect what-if snapshot with the historical snapshot, as the current topology may change and what-if snapshot was created based on the historical context

* how to connect to the current network instance

* how to model and retrieve potential network: either full topology or only changes from the snapshot

* how to retrieve the external models via links in the what-if context

* is this SIMAP requirement or generic requirement for any YANG retrieval via NETCONF/RESTCONF. The [RFC8342] defines different datastores: configuration datastores (startup, candidate, running, intended) and operational datastore. It can potentially be used for one candidate, but it does not support multiple candidates with links to different running. We have private candidates that can be potentially used.

Please refer to Section 6 for analysis of multiple endpoints (sematics determined by the selection of the endpoint) versus single endpoint (semantics in the model).

7.4.3.2. Implementation Proposal

The following is the initial proposal:

* provide a solution that would allow the live versus potential to be modeled via different endpoints(please see Section 6)

* optionally, provide the solution for adding the semantics to the model:

  - use list network for potential as well

  - network-id is generated for each potential candidate

  - additional info needed: category (potential), timestamp, relation to the live network network-id

  - relation to historical snapshot may be determined based on the live network-id and timestamp

Preferably, the generic solution with private candidates would be used, but for now we have the following modelled.

```
  identity network-category {
    description "Base identity for the network category";
  }

  identity network-category-none {
    description "This is used when endpoint selection identifies if
                 live, potential or intended. This is the default";
  }

  identity network-category-potential {
    base "st:network-category";
    description "Potential topology. There may be optionally one or
                 multiple instances of the potential network, related
                 to the current network. Timestamp determines what
                 historical snapshot has been used when
                 creating the potential network instance";
  }

  augment "/nw:networks/nw:network" {

    if-feature simap-lifecycle;
    description
      "Adding optional capability for modelling of potential
       and intended network";

    leaf network-category {
      type identityref {
        base st:network-category;
      }
```

```
         description
           "The network category: none, live, potential, intended.
            Default is none.";
       }
       leaf live-network-ref {
         description
           "Required for potential and intended only to connect
            to the live network instance";
          type leafref {
            path "/nw:networks/nw:network/nw:network-id";
          require-instance false;
       }
       leaf timestamp {
         description
           "Required for potential only to connect to the
            historical snapshot";
         type yang:date-and-time;
       }
     }
```

Figure 12: The proposal for potential snapshots

Check if this proposal has an issue with backward compatibility, as the read would return potential instances as well. Alternative is to have a separate list networks-potential.

7.4.4. REQ-INTENDED: Intended topology

7.4.4.1. Analysis

SIMAP must enable both retrieval and write access to the intended network topology that cannot be discovered from the real network (for example target L2 Topology, L3 Topology, passive topology that cannot be discovered).

This requires the implementation proposal to take into account the following:

* how to connect intended to the current network instance

Please refer to Section 6 for analysis of multiple endpoints (sematics determined by the selection of the endpoint) versus single endpoint (semantics in the model).

Preferably we would use NMDA and one of the datastores if possible, needs further analysis.

7.4.4.2. Implementation Proposal

The following is the initial proposal:

* provide a solution that would allow the live versus intended to be modeled via different endpoint (please see Section 6)

* optionally, provide the solution for adding the semantics to the model:

  - use list network for intended as well

  - network-id is generated for intended network, different network-id from the live network

  - additional info needed: type: intended, relation to the live network network-id

Preferably, the generic solution with standard datastored would be used if possible, but for now we have the following modelled.

```
  identity network-category-none {
    description "This is used when endpoint selection identifies if
                 live, potential or intended. This is the default";
  }

  identity network-category-intended {
    base "st:network-category";
    description "Intended topology, there is optionally 1 instance of
                 intended network related to the live network instance";
  }

  augment "/nw:networks/nw:network" {
    if-feature simap-lifecycle;

    description
      "Adding optional capability for modelling of potential and
      intended network";

    leaf network-category {
      type identityref {
        base st:network-category;
      }
      description
        "The network category: live, potential, intended.
        Default is live.";
    }
    leaf live-network-ref {
      description
        "Required for potential and intended only to connect to the
        live network instance";
       type leafref {
         path "/nw:networks/nw:network/nw:network-id";
       require-instance false;
       }
    }
    :
    :
  }
```

Figure 13: The proposal for intended snapshots

Check if this proposal has an issue with backward compatibility, as the read would return potential instances as well. Alternative is to have a separate list networks-intended.

7.4.5. REQ-STATUS: Links and nodes down in topology

7.4.5.1. Analysis

Links and nodes that are down must appear in the topology. The status of the nodes and links must be either implemented in the SIMAP or accessible from the SIMAP. Whether the status is included as part of
the SIMAP or accessible from the SIMAP is left to the solutions.

Therefore, based on this requirement we must optionally support the status in SIMAP.

7.4.5.2. Implementation Proposal

Options for showing links in the topology: * rely on the intended network for the target and the live network
only shows the current topology * real network should include both intended network and the state that reflects the current status

This draft proposes to use intended for the target state and that application can retrieve live and intended and compute the diff and merge them if needed.

The draft also proposes to add status, if required, via REQ-EXTENSIBLE in Section 7.2.2.

7.4.6. REQ-TEMPO-HISTORY: Geo-spatial, temporal, historical

7.4.6.1. Analysis

Move any relevant text from draft-havel-nmop-digital-map.

TODO: Do we need to add timestamps here for temporal.

Proposal for historical is done in REQ-SNAPSHOT.

In regards to geo-spatial, there are 2 options: * refer to existing YANG module * implement here This draft proposes to link to the external

7.4.6.2. Implementation Proposal

Proposal for historical is done via REQ-SNAPSHOT Proposal for geo-spacial is done via REQ-PLUGG TODO: Proposal for temporal requires further analysis

7.5. Feature simap-external

The solution is proposed for the following requirements: * REQ-STD-API-BASED * REQ-PLUGG

7.5.1. REQ-STD-API-BASED: Standard based

7.5.1.1. Analysis

RFC8345 already supports the standard model and API for the SIMAP requirements supported by RFC8345. We also need a standard model and API for the SIMAP requirements identified as RFC8345 gaps. This requirement also mentions that these APIs must also provide the capability to retrieve the links to external data/models.

7.5.1.2. Implementation Proposal

The implementation for the model changes for this requirement will be covered by the implementation proposal for REQ-PLUGG in Section 7.5.2.

7.5.2. REQ-PLUGG: Pluggable

7.5.2.1. Analysis

This requirement states that SIMAP must be pluggable and connect to other YANG modules and other nodelling mechanisms.

7.5.2.2. Implementation Proposal

The solution for the external links is proposed in a separate draft [I-D.vivek-simap-external-relationship].

TODO - Check if there is alternative proposal with some simple references

8. Model Structure Details

The SIMAP data model is defined in the "ietf-simap-topology" module. Its structure is shown in Figure 1. The notation syntax follows the syntax used in [RFC8340].

```
module: ietf-simap-topology
  +--ro networks-history
     +--ro network-history* [live-network-ref timestamp]
        +--ro live-network-ref    -> /nw:networks/network/network-id
        +--ro timestamp           yang:date-and-time
        +--ro file-id?            inet:uri

  augment /nw:networks/nw:network:
    +--rw name?          string
    +--rw label*         string
    +--rw description?   string
    +--ro extension?     <anydata>
  augment /nw:networks/nw:network/nw:node:
    +--rw name?          string
    +--rw label*         string
    +--rw description?   string
    +--ro extension?     <anydata>
  augment /nw:networks/nw:network/nw:node/nt:termination-point:
    +--rw name?          string
    +--rw label*         string
    +--rw description?   string
    +--ro extension?     <anydata>
  augment /nw:networks/nw:network/nt:link:
    +--rw name?          string
    +--rw label*         string
    +--rw description?   string
    +--ro extension?     <anydata>
  augment /nw:networks/nw:network:
    +--rw network-category?   identityref
    +--rw live-network-ref?   -> /nw:networks/network/network-id
    +--rw timestamp?          yang:date-and-time
  augment /nw:networks/nw:network/nt:link:
    +--rw link-type?        identityref
    +--rw link-direction?   identityref
    +--rw tp* [network-ref node-ref tp-ref]
       +--rw tp-ref          -> /nw:networks/
network[nw:network-id=current()/../network-ref]/
node[nw:node-id=current()/../node-ref]/nt:termination-point/tp-id
       +--rw node-ref        -> /nw:networks/
network[nw:network-id=current()/../network-ref]/node/node-id
       +--rw network-ref     -> /nw:networks/network/network-id
       +--rw tp-role?        identityref
       +--rw tp-direction?   identityref
  augment /nw:networks/nw:network:
    +--rw subnetwork* [network-ref]
       +--rw network-ref    -> /nw:networks/network/network-id
```

Figure 14: The Structure of the SIMAP Data Model

9. YANG Module

```
module ietf-simap-topology {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-simap-topology";
  prefix st;

  import ietf-network {
    prefix nw;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  import ietf-network-topology {
    prefix nt;
    reference
      "RFC 8345: A YANG Data Model for Network Topologies";
  }

  organization
    "Network Management Operations (NMOP) Working Group";

  contact
    "WG Web:    <https://datatracker.ietf.org/group/nmop/>
     WG List:   <mailto:nmop@ietf.org>

     Editor:    Olga Havel
                <mailto:havel.olga@huawei.com>

     Editor:    Nigel Davis
                <mailto:ndavis@ciena.com>

     Editor:    TODO
                <mailto:TODO>";


  description
    "This module defines the SIMAP core topology model based on the
     requirements defined in
     https://datatracker.ietf.org/doc/draft-ietf-nmop-simap-concept/.

     Copyright (c) 2025 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust’s Legal Provisions
```

```
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
     (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
     for full legal notices.

    ";

  revision 2025-01-01 {
    description
      "Initial revision.";
    reference
      "XXX: A YANG Data Model for SIMAP Core Topologies";
  }



  identity tp-role {
    description "Base identity from which all tp roles in the link are
                 derived.";
  }

  /*
   * Common SIMAP groupings for optional RFC8345 extensions
   * Addressing requirements REQ-PROPERTIES as name, label and
   * description may be important properties that
   * clarify the topological roles for different layers and technologies
   */
  grouping simap-common {
    description "A reusable set of optional extensions for network,
                 node, termination point and link";
    leaf name {
      type string;
      description
        "The user friendly name, if required.
        It is optional, for backward compatibility";
    }
    leaf-list label {
      type string;
      description
        "Used for optionally adding any labels to the instances,
         if required";
    }
    leaf description {
      type string;
      description
        "Used for optionally adding any description to the instances,
```

```
          if required";
      }
      anydata extension {
        config false;
        description
          "The extension point for any other meta info or data or any
          unknown extensions. Proposed solution for SIMAP requirement
          REQ-EXTENSIBLE. Any additional info that is topologically
          significant can also be added this way for requirement
          REQ-TOPO-ONLY or REQ-PROPERTIES.";
      }
    }

    augment "/nw:networks/nw:network" {
      description
        "Adding optional common simap extensions";
      uses st:simap-common;
    }

    augment "/nw:networks/nw:network/nw:node" {
      description
        "Adding optional common simap extensions";
      uses st:simap-common;
    }

    augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
      description
        "Adding optional common simap extensions";
      uses st:simap-common;
    }

    augment "/nw:networks/nw:network/nt:link" {
      description
        "Adding optional common simap extensions";
      uses st:simap-common;
    }

    /*
     * Candidate for potential (REQ-POTENTIAL) and intended (REQ-INTENDED)
     * topology requirement
     */

     /*
     * Identities related to type of the network:
     *  live, potential, intended
     */
    identity network-category {
      description "Base identity for the network category";
```

```
  }

  identity network-category-none {
    description "This is used when endpoint selection identifies if
                 live, potential or intended. This is the default";
  }

  identity network-category-live {
    base "st:network-category";
    description "Live topology, there is only 1 instance of the live
                 network";
  }

  identity network-category-potential {
    base "st:network-category";
    description "Potential topology. There may be optionally one or
                 multiple instances of the potential network,
                 related to the current network";
  }

  identity network-category-intended {
    base "st:network-category";
    description "Intended topology, there is optionally 1 instance of
                 intended network related to the live network instance";
  }

  augment "/nw:networks/nw:network" {

    description
      "Adding optional capability for modelling of potential and
      intended network";

    leaf network-category {
      type identityref {
        base st:network-category;
      }
      description
        "The network category: live, potential, intended.
        Default is live.";
    }
    leaf live-network-ref {
      description
        "Required for potential and intended only to connect to the
        live network instance";
       type leafref {
         path "/nw:networks/nw:network/nw:network-id";
       require-instance false;
       }
```

```
    }
    leaf timestamp {
      description
        "Required for potential only to find related  historical
        snapshot";
      type yang:date-and-time;
    }
  }

  /*
   * Adding bidirectional and multi-point capabilities to the link
   */

  /*
   * Identities related to link directionality and cardinality of points
   */

  identity link-direction {
    description "Base identity for the directionality of the link";
  }

  identity link-direction-uni {
    base "st:link-direction";
    description "Unidirectional link";
  }

  identity link-direction-bi {
    base "st:link-direction";
    description "Bidirectional link";
  }

  /*
   * Identities related to link type and cardinality of points
   */
  identity link-type {
      description
        "Base identity for the internal structure of the link";
  }

  identity link-type-p2p {
      base "st:link-type";
      description "Point to point link";
  }

  identity link-type-p2mp {
      base "st:link-type";
      description "Point to multi-point link";
  }
```

```
  identity link-type-mp2mp {
      base "st:link-type";
      description "Multi-point to multi-point link";
  }

  identity link-type-hybrid {
      base "st:link-type";
      description
        "Hybrid links, combination of the other three"
        The multi-point link is an abstraction of underlying structure
        detail. The underlying structure is currently expressed in
        terms of a simple identity which essentially references
        a pattern of underlying flow where that pattern describes the
        effect (another abstraction) of that flow between each point
        of the link. There are currently a few well understood
        patterns, but the list will expand. The list of patterns, not
        surprisingly, will be the same as for  service, for
        connectivity and for flow (as the link arises from them). The
        approach of introduction of hybrid links allows for a
        solution to express a complex arrangement in abstraction without
        needing, for each instance, to lay out the detail. It also
        allows the client application to “understand” the effect
        without having to probe that detail.";
  }

  /*
   * Identities related to termination point directionality and role
   * in the link
   */

  identity tp-direction {
    description "Base identity for the directionality of the tp";
  }

  identity tp-direction-symmetric {
    base "st:tp-direction";
    description "TP in the bidirectional link";
  }

  identity tp-direction-source {
    base "st:tp-direction";
    description "TP is the source tp in the link";
  }

  identity tp-direction-destination {
    base "st:tp-direction";
    description "TP is the destination tp in the link";
  }
```

```
  augment "/nw:networks/nw:network/nt:link" {
    description
      "Models the link that can be unidirectional, bidirectional,
      point-to-point,point-to-multi-point, multipoint-to-multipoint,
      therefore addressing requirements:
      - REQ-BIDIR
      - REQ-MULTI-POINT

      with augmenting we keep source and destination for backward
      compatibility.

      Allows for the link to terminate on the termination point that is
      on another network, therefore also addressing the gap:
      - REQ-MULTI-DOMAIN
      ";

    leaf link-type {
      type identityref {
        base st:link-type;
      }
      description
        "The reference to the specification for the internal structure
        of the link. It can be point to point (link-type-p2p), point to
        multipoint (link-type-p2mp) or multipoint to multipoint
        (link-type-mp2mp). It can also be any other custom identity
        defined with base link-type. It is optional, so the model
        supports the solution without the link type information,
        either if not known or for backward compatible case.";
    }

    leaf link-direction {
      type identityref {
        base st:link-direction;
      }
      description
        "The direction of the link: unidirectional (link-direction-uni)
        or bidirectional (link-direction-bi). It can also be any other
        custom identity defined with base link-direction. It is
        optional, so the model supports the solution without the link
        direction information, either if not known or for backward
        compatible case.";
    }

    list tp {
      key "network-ref node-ref tp-ref";
      description "List of termination points in the link";
      uses nt:tp-ref;
```

```
      leaf tp-role {
        type identityref {
          base st:tp-role;
        }
        description
          "The role of the termination point in the link defined in the
           link-type spec.";
        }
      leaf tp-direction {
        type identityref {
          base st:tp-direction;
        }
        description
          "The direction of the point in the link";
      }
    }
  }

  /*
   * Subnetworks, requirement REQ-SUBNETWORK
   */
  augment "/nw:networks/nw:network" {
    list subnetwork {
        key "network-ref";
           description
             "A subnetwork of the network, supports partitioning";
           leaf network-ref {
             type leafref {
               path "/nw:networks/nw:network/nw:network-id";
             require-instance false;
             }
             description
               "References the subnetworks";
           }
    }
  }

  /*
   * Candidate for graph traversal (REQ-GRAPH-TRAVERSAL)
   * topology requirement (tp->tp, node->node)
   */
  augment "/nw:networks/nw:network/nw:node/nt:termination-point" {
    description
      "Adding optional read only link relations between connected
termination points for optimized graph traversal for paths";

    list linked-termination-point {
      config false;
```

```
      key "network-ref node-ref tp-ref";
      description
            "This list identifies any linked termination points, added
             optionally for read only for optimized path graph
             traversal";

      uses nt:tp-ref;
  }

  augment "/nw:networks/nw:network/nw:node" {
    description
      "Adding optional read only link relations between connected
nodes for optimized graph traversal for paths";

    list linked-node {
      config false;
      key "network-ref node-ref";
      description
            "This list identifies any linked nodes, added
             optionally for read only for optimized path graph
             traversal";

      uses nw:node-ref;
  }

  /*
   * Networks history, requirement REQ-SNAPSHOT
   */

  container networks-history {
    config false;
    list network-history {
      key "live-network-ref timestamp";
      description
        "Historical network. Snapshot is generated for each historical
        instance, network-ref is the reference for the live network";
      leaf live-network-ref {
         type leafref {
           path "/nw:networks/nw:network/nw:network-id";
         }
      }
      leaf timestamp {
        type yang:date-and-time;
      }
      leaf file-id {
        type inet:uri;
      }
    }
```

```
  }
}
```

Figure 15: The YANG Data Model for SIMAP

10. Security Considerations

11. IANA Considerations

This document has no actions for IANA.

12. References

12.1. Normative References

[I-D.ietf-nmop-simap-concept]
Havel, O., Claise, B., de Dios, O. G., and T. Graf, "SIMAP: Concept, Requirements, and Use Cases", Work in Progress, Internet-Draft, draft-ietf-nmop-simap-concept-09, 18 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-simap-concept-09>.

[I-D.vivek-simap-external-relationship]
Boudia, V., Francois, P., and S. Mostafa, "External Relationship model for SIMAP", Work in Progress, Internet-Draft, draft-vivek-simap-external-relationship-00, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-vivek-simap-external-relationship-00>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/rfc/rfc8040>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/rfc/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/rfc/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/rfc/rfc8342>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/rfc/rfc8345>.

[RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for the Routing Information Base (RIB)", RFC 8431, DOI 10.17487/RFC8431, September 2018, <https://www.rfc-editor.org/rfc/rfc8431>.

12.2. Informative References

[RFC8795] Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Gonzalez de Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", RFC 8795, DOI 10.17487/RFC8795, August 2020, <https://www.rfc-editor.org/rfc/rfc8795>.

Acknowledgments

Authors’ Addresses

Olga Havel
Huawei
Email: olga.havel@huawei.com

Nigel Davis
Ciena
Email: ndavis@ciena.com

Benoit Claise
Huawei
Email: benoit.claise@huawei.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com

Thomas Graf
Swisscom
Email: thomas.graf@swisscom.com