Network Management Operations
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

O. Havel
Huawei
N. Davis
Ciena
B. Claise
Huawei
O. G. D. Dios
Telefonica
T. Graf
Swisscom
7 July 2025

A YANG Data Model for SIMAP
draft-havel-nmop-simap-yang-00

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/ data (e.g., inventory, observability sources, and 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 peroposes the solution how to implement these gaps.

2. Terminology

2.1. Why RFC8345 is a Good Approach for SIMAP Modelling

The main reason for selecting RFC8345 for modelling is its simplicity and that is supports either fully or partially the majority of the core requirements.

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

* generic requirements for all YANG modules,

* requirements already supported by [RFC8345],

* requirements identified as [RFC8345] gaps, they require extensions/changes to ietf-network and ietf-network-topology YANG modules.

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

2.2. Generic requirements

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

* Architectural 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.

   - 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

2.3. 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.

   - 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.

2.4. 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.

  - REQ-STD-API-BASED: Standard based SIMAP models and APIs, for multi-vendor support. Gap: links are entities, adding linkedTo relations would help

  - REQ-GRAPH-TRAVERSAL: Graph Traversal

  - REQ-SNAPSHOT: Network snapshot topology.

  - REQ-POTENTIAL: Potential new network topology.

  - REQ-INTENDED: Intended topology.

  - REQ-SEMANTIC: Network topology semantics. Gap: some semantic missing.

  - REQ-EXTENSIBLE: Extensible via metadata

  - REQ-PLUG: SIMAP must be pluggable

  - REQ-BIDIR: SIMAP must provide a mechanism to model bidirectional links. Gap: complex

  - REQ-MULTI-POINT: SIMAP must provide a mechanism to model multipoint links. Gap: complex

  - REQ-MULTI-DOMAIN: SIMAP must provide a mechanism to model links between networks

  - REQ-SUBNETWORK: SIMAP must provide a mechanism to model network decomposition into sub-networks

  - 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

  - 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

* 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.

  - REQ-RELATIONSHIPS: SIMAP should contain all topological relationships inside each layer or between the layers (underlay/overlay)

  - REQ-TEMPO-HISTO: Must support geo-spatial, temporal, and historical data.

2.5. 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.

- 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.

2.6. Requirements for further analysis

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

* Operator Requirements:

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

  - REQ-SHARED: Share nodes, links, and termination points between different networks.

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.

3. 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.

4. 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 we have separate API endpoints for the live network and potential networks. These are the consequences if we follow this approach:

* the same topology model can be used between the live, potential and intended network

* we have different API endpoints for the 3 categories

* the access rights (read for live and read/write for potential) could be done outside of the model (NACM control access rules [RFC8341]), so we don’t need to use config false/true.

* we delegate the tracking of relations between the potential, live network, intended network and historical snapshots to the applications and the applications needs to orchestrate between different APIs to get the diff from the snapshot

In order to enable more advanced solutions and support positioning of SIMAP at different interfaces (Chapter 5 in [I-D.ietf-nmop-simap-concept]), the current version of the draft proposes the following:

* have a default behavior that allows for separation between live, potential and intended, without having that part of semantics defined in the model (e.g. semantics is determined by the selection of the endpoint and the model does not specify the category of the topology).

* enable unified API as optional, introduce the model with semantics identifying what is live, potential and intended topology, that models the relations between the live, potential, historical and intended instances. This can be used by advanced solutions that require model to include this part of semantics.

5. Solution Proposal for the RFC8345 Gaps

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

5.1.1. 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, etc).

5.1.2. Implementation Proposal

The implementation for the model changes for this requirement will be covered by the implementation proposal for REQ-POTENTIAL in Section 6.5.

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

5.2.1. 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.

5.2.2. Implementation Proposal

The implementation for the model changes for this requirement will be covered by the implementation proposal for REQ-PLUGG in Section 6.9.

5.3. REQ-GRAPH-TRAVERSAL: Graph Traversal

5.3.1. 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 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.

5.3.2. 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";

      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";

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

        uses nw:node-ref;
    }
```

Figure 1: The proposal for optimized path graph traversal

5.4. REQ-SNAPSHOT: Different snapshots

5.4.1. 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.

5.4.2. Implementation Proposal

The following are the identified implementation candidates:

* this is generic requirement, not just for SIMAP, therefore implementation should not be implemented by SIMAP but by the NETCONF/RESTCONF for historical system state + conf

* 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, currently proposed in the YANG module:

```
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 2: The proposal 1 for historical snapshots

* implemented as a 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 {
  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 3: The proposal 2 for historical snapshots

The proposal 1 with file reference is currently implemented in the YANG module.

5.5. REQ-POTENTIAL: Potential new network topology

5.5.1. 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.

Please refer to Chapter 5: API Scope for analysis of multiple endpoints (sematics determined by the selection of the endpoint) versus single endpoint (semantics in the model).

5.5.2. Implementation Proposal

The following is the initial proposal:

* provide a solution that would allow the live versus potential to be modeled via different endpoint (please see Chapter 5: API Scope)

* 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

```
     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" {

       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 4: 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.

5.6. REQ-INTENDED: Intended topology

5.6.1. 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, etc).

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

* how to connect intended to the current network instance

Please refer to Chapter 5: API Scope for analysis of multiple endpoints (sematics determined by the selection of the endpoint) versus single endpoint (semantics in the model).

5.6.2. 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 Chapter 5: API Scope)

* 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

```
  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" {

    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 5: 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.

5.7. REQ-SEMANTIC: Network Topology Semantic

5.7.1. Analysis

5.7.2. Implementation Proposal

5.8. REQ-EXTENSIBLE: Extensible via metadata

5.8.1. 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.

5.8.2. Implementation Proposal

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

```
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.";
}
```

Figure 6: The proposal for extensions for meta data

5.9. REQ-PLUGG: Pluggable

5.9.1. Analysis

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

5.9.2. Implementation Proposal

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

5.10. REQ-BIDIR: Bidirectional Links

5.10.1. 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, etc. We also encounter 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. 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 the bidirectional links

5.10.2. 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 {
       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-direction {
         type identityref {
           base st:tp-direction;
         }
         description
           "The direction of the point in the link";
       }
   }
```

Figure 7: The proposal for bidirectional links

5.11. REQ-MULTI-POINT: Multipoint Links

5.11.1. 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, complex).
  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. 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

5.11.2. 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";
   }

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

     :
     :
     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.";
     }
     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.";
         }
     :
     :
     }

   }
```

Figure 8: The proposal for multi-point links

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

5.12.1. 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

5.12.2. 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 {
    key "network-ref node-ref tp-ref";

  }
}
```

Figure 9: The proposal for multi-domain links

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

5.13. REQ-SUBNETWORK: Subnetworks and partitioning

5.13.1. 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.

5.13.2. Implementation Proposal

```
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";
         }
  }
}
```

Figure 10

5.14. REQ-SUPPORTING: Extension to supporting

5.14.1. Analysis

Move any relevant text from draft draft-havel-nmop-digital-map about the current RFC8345 solution for supporting and why this is not the right solution.

5.14.2. Implementation Proposal

Propose the implementation.

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

5.15.1. 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.

5.15.2. Implementation Proposal

Options: * rely on the intended network for the target and the live network only shows the current state * real network should include both intended and state that reflects the 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.

5.16. REQ-PROPERTIES: Properties significant for topology

5.16.1. 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.

5.16.2. 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 {
      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 {
    :
    :
  }
```

Figure 11: The proposal for bidirectional links

5.17. REQ-RELATIONSHIPS: Relationships significant for topology

5.17.1. Analysis

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

5.17.2. Implementation Proposal

Propose the implementation.

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

5.18.1. Analysis

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

Analyze temporal (same table) versus historical (separate tables), temporal may not be needed here. 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

5.18.2. Implementation Proposal

Propose the implementation

6. 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 12: The Structure of the SIMAP Data Model

7. 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";
  }

  /*
   * 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";
           }
    }
  }

  /*
   * 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 13: The YANG Data Model for SIMAP

8. Security Considerations

9. IANA Considerations

This document has no actions for IANA.

10. 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-05, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-simap-concept-05>.

[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>.

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