Network Management Operations
Internet-Draft
Intended status: Informational
Expires: 5 December 2026

O. Havel
Huawei
B. Claise
Everything OPS
O. G. D. Dios
Telefonica
T. Graf
Swisscom

3 June 2026

SIMAP: Concept, Requirements, and Use Cases
draft-ietf-nmop-simap-concept-11

Abstract

This document defines the concept of Service & Infrastructure Maps (SIMAP) and identifies a set of SIMAP requirements and use cases. The SIMAP was previously known as Digital Map. SIMAP evolves the earlier ’Digital Map’ concept by making explicit the ties between service and infrastructure layers, clarifying expected outcomes for operations and automation, and addressing ambiguity associated with the term ’digital.’

The document intends to be used as a reference for the assessment of the various topology modules to meet SIMAP requirements.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Network Management Operations Working Group mailing list (nmop@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/nmop/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-wg-nmop/draft-ietf-nmop-digital-map-concept.

Status of This Memo

Table of Contents

1. Introduction

This document defines the concept of Service & Infrastructure Maps (SIMAP) and outlines associated requirements and use cases.

SIMAP is a data model that provides a topological view of the operator's networks and services, including how it is connected to other models (e.g., inventory) and external data sources (e.g., observability data, and operational knowledge). This model represents a multi-layered topology and offers mechanisms to navigate amongst layers and correlate between them, including layers from physical to service topology. This model is applicable to multiple domains (access, core, data center, etc.) and technologies (Optical, IP, etc.). While this document refers to SIMAP as a data model to reflect the Working Group's intent that it be concretely implementable, the actual data model specification — including the choice of modelling language and implementation approach — is out of scope of this document and will be defined in companion specifications.

Specifically, the SIMAP modelling defines the core topological entities at each layer, core topological properties, and topological relationships (both inside each layer and between the layers), to ensure a multi-layered topology can be reconstructed, validated and queried in an unambiguous and interoperable manner. The core topological entities are the minimal set of objects required to represent a layer's topology (e.g., network, node, termination point, and link). The core topological properties are the essential attributes associated with these entities (e.g., identity, topology type, entity role in topology, directionality, cardinality, and cost/weight), enabling analysis of how the network structure affects services and operations. For example, topological reasoning can answer questions such as: 'If link X fails, what services are impacted?' or 'What is the full end-to-end data path of the service flow?'.

The additional concepts or attributes (such as capacity, operational state, performance metrics, or inventory data) are modelled outside of SIMAP, the core set provides the necessary structure to support these extensions without losing architectural consistency.

The SIMAP modelling also defines how to access other external models from a topology. SIMAP is a topological model that is linked to other functional models and connects them all: configuration, maintenance, assurance (KPIs, status, health, and symptoms), Traffic-Engineering (TE), different behaviors and actions, simulation, emulation, mathematical abstractions, AI algorithms, etc. These other models exist outside of the SIMAP and are not defined during SIMAP modelling.

The SIMAP data consists of instances of network and service topologies at different layers. There may be a separate topology instance for each layer in a multi-layered network, or a single topology instance that encompasses multiple layers. Since SIMAP is a data model and data models can generate APIs [RFC3444][RFC7950], the SIMAP provides access to this data via standard APIs for both read and write access, typically from a controller, with query capabilities and links to other data models (e.g., Service Assurance for Intent-based Networking (SAIN) [RFC9417], Service Attachment Points (SAPs) [RFC9408], Inventory [I-D.ietf-ivy-network-inventory-yang], and potentially linking to non-YANG models).

The SIMAP also provides write operations with the same set of APIs, not to change a topology layer on the fly as a northbound interface from the controller, but for both online and offline simulations, before applying the changes to the network via the normal controller operations.

Both real network, online simulation, and offline simulation APIs can be built on the same data model. The real network API reflects actual changes in the topology as reported by the SIMAP server. Online simulation applies hypothetical changes to the current live model to assess immediate impacts (e.g., if link X fails, what services are disrupted), without altering the real network. Offline simulation applies hypothetical changes to a saved or alternate model, useful for planning, training, or evaluating changes before deployment. Each data source is reported as a distinct topology instance, but when desired the real network and online simulation data can be merged into a single topology instance, while the offline simulation remains separate. The simulated topology instance can be matched directly to the corresponding real network topology for comparison. This approach preserves independence between real and simulated data while enabling side by side analysis.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The normative keywords in this document define requirements for future SIMAP specifications and implementations that claim conformance to SIMAP, and do not impose requirements on this document itself.

This document makes use of the following terms:

Domain: A collection of network resources, services, and management functions that are administered under a common set of policies or operational control. Domains may correspond to technology boundaries (e.g., optical, IP), functional areas (e.g., access, core, data center), or administrative partitions. Multi-domain SIMAP operations involve correlating topology and service information across such domains.

Topology: Topology refers to the network and service topology. A network topology defines how physical or logical nodes, links and termination points are related and arranged. A Service topology defines how service components (e.g., VPN instances, customer interfaces, and customer links) between customer sites are interrelated and arranged.

There are several types of topologies: point-to-point, bus, ring, star, tree, mesh, hybrid, and daisy chain.

Topologies may be unidirectional (all links unidirectional) or bidirectional (all links bidirectional), or contain combination of unidirectional and bidirectional links.

Multi-layered topology: A multi-layered topology models relationships between different topology layers, where each layer represents a connectivity aspect of the network and services that needs to be configured, controlled and monitored. Each topology layer has a separate lifecycle.

[RFC8345] also refers to this multi-layered topology as topology

hierarchy (stack). It also uses layers when describing supporting relations (represent layered network topologies), underlay/ overlay, network nodes and layering information. [RFC8345] states that the model can be used for representation of layered network topologies.

[RFC8345] is flexible and can support both the same network topology instance with multiple layers (e.g., Layer 2 and Layer 3) or separate network topology instances with supporting relations between them (e.g., separate Layer 2 and Layer 3). Therefore, multiple topology layers can be grouped into the same network topology instance, if solution requires.

Topology layer: A topology layer represents Topology at a single layer in the multi-layered topology.

The topology layer can also represent what needs to be managed by a specific user or client application, for example the IGP layer can be of interest to the operator troubleshooting or optimizing the routing, while the optical layer may be of interest to the user managing the optical network.

Some topology layers may relate closely to OSI layers, like Layer 1 topology for physical topology, Layer 2 for link topology and Layer 3 for IPv4 and IPv6 topologies.

Some topology layers represent the network topology of Layer 3 control protocols, like OSPF, IS-IS, or BGP.

The service layer represents the Service view of the connectivity, that can differ for different types of Services and for different providers/solutions.

The application/flow layer represents the view of Service data flows for different classes of service - video, voice and data traffic. The layers may differ depending on the solution, so the bottom and top layers may not be the same across all solutions. We can illustrate the concept of topology layers by listing the common set - e.g.,

* physical: L1, one or multiple layers, if fully modelling different optical layers. Used for WSON, OTN optical, OTN digital),

* data link: L2 for Ethernet, LAGs and VLAN,

* network: L3 for IPv4 and IPv6,

* IGP/EGP: for routing inside or between ASs, different layers for underlay and overlay, for ISIS, OSPF, iBGP. eBGP,

* tunnel: for transport tunnels and paths, for MPLS and SRv6,

* service: for different overlay services, like L2 VPNs, L3 VPNs, slices, SD-WAN,

* application: for video, voice and data traffic flows.

However, this list is illustrative only; it is not a prescriptive requirement. Different solutions may adopt alternative layering schemes or combine layers differently. Therefore, we will present the above as an example of one possible solution, while keeping the definition flexible enough to accommodate flexible layering.

Service: A service represents network connectivity service provided over a network that enables devices, systems, or networks to communicate and exchange data with each other. It provides the underlying infrastructure and mechanisms necessary for establishing, maintaining, and managing connections between different endpoints. The example services are: L2VPN, L3VPN, EVPN, VPLS, VPWS,

Resource: Defined in [I-D.ietf-nmop-terminology]

Termination Point: Defined in [RFC8345], as follows:

The network-topology module defines a topology graph and components from which it is composed: nodes, edges, and termination points. Nodes (from the "ietf-network" module) represent graph vertices and links represent graph edges. Nodes also contain termination points that anchor the links.

A node has a list of termination points that are used to terminate links. An example of a termination point might be a physical or logical port or, more generally, an interface. Like a node, a termination point can in turn be supported by an underlying termination point, contained in the supporting node of the underlay network.

The document defines the following terms:

Service & Infrastructure Maps (SIMAP): SIMAP is a data model that provides a topological view of the operator’s networks and services, including how it is connected to other models (e.g., inventory) and external data sources (e.g., observability data, 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 centers, etc.) and technologies (Optical, IP, etc.)

Therefore, SIMAP defines the core topological entities, their role in the network, core topological properties, and relationships both inside each layer and between the layers. It is a basic topological model with references/pointers to other models and connects them all: configuration, maintenance, assurance (KPIs, status, health, symptoms, etc.), traffic engineering, different behaviors, simulation, emulation, mathematical abstractions, AI algorithms, etc.

SIMAP modelling: SIMAP modelling is the set of principles, guidelines, and conventions to model the SIMAP. They cover the network types (layers and sublayers), entity types, entity roles (network, node, termination point, or link), entity properties, relationship types between entities and relationships to other entities.

SIMAP data: SIMAP data consists of instances of network and Service topologies at different layers. This includes instances of networks, nodes, links and termination points, topological relationships between nodes, links and termination points inside a network, relationships between instances belonging to different networks, links to other non-topological data for the instances (e.g., inventory, configuration, health, symptoms).

The SIMAP data can be historical, real-time, or future data for 'what-if' scenarios.

SIMAP API: SIMAP API is the set of interfaces that allow the client applications to create, read, update, and delete data that comforms to the SIMAP.

SIMAP client application: Consumer of the SIMAP API. An application that consumes the SIMAP API. It sends requests to a SIMAP server, receives responses, and uses the returned data to drive its own logic. Typical clients include network management systems, orchestration tools, monitoring dashboards, capacity management applications, or any software that needs to read or modify the topology information defined by the SIMAP. The client is responsible for forming valid API calls, handling authentication/ authorization, parsing responses, and translating the SIMAP into its own internal representation.

SIMAP server: Provider of the SIMAP API. An application or a system that implements the API endpoints to expose the SIMAP data model to external consumers, building it from live network state or simulation scenarios. The server accepts requests to create, read, update, delete, or query instances of the SIMAP topology, validates input against the data model schema, persists changes (if any), and returns responses that conform to the SIMAP API specification. The server's implementation may reside inside a controller, orchestrator, device, service manager, or any other application/system-or be a standalone application/system-depending on the solution architecture. The server may offer ancillary services such as authentication, rate limiting, versioning, logging, and monitoring, but its primary role is to expose the SIMAP via programmable interface.

3. Sample SIMAP Use Cases

The following subsections provide a non-exhaustive list of SIMAP use cases, with a focus on the related SIMAP client application requirements and its interactions with SIMAP server, in order to extract the SIMAP-related requirements (Section 4).

3.1. Common Enablers for SIMAP

This section identifies a set of enablers that are invoked when providing the various business-oriented SIMAP use cases. These enablers are grouped here to avoid duplication.

3.1.1. Service -> Resource

The SIMAP APIs can be invoked to retrieve all Services for selected service types. A SIMAP client application that triggers such a request will be able to retrieve the topology for selected Services via the SIMAP APIs and, from the response, it will be able to navigate top-down to the lower layers via the supporting relationship provided by the SIMAP server. In doing so, the SIMAP client application will be able to determine what logical resources are used by a Service. The supporting relations to the lowest layer, provided by the SIMAP server, will help the SIMAP client application to determine what physical resources are used by the Service. This addresses a requirement for systems to be able to provide topology and resource views of services, at different levels of abstraction, using the SIMAP [ETSI-ZSM-019]. Knowing the physical resources a service uses enables capacity planning, fault isolation, performance monitoring, and accurate billing.

3.1.2. Resource -> Service

A SIMAP client application can navigate from the physical, Layer 2, or Layer 3 topology to the Services that rely upon specific resources. For example, the application will be able to select the resources and by navigating the supporting relationship bottom-up come to the Service and its nodes, termination points and links.

These APIs can be invoked for Service impact analysis, for example.

3.1.3. Traffic Engineering (TE)

Traffic Engineering (TE) [RFC9522] is a network optimization technique designed to enhance network performance and resource utilization by intelligently controlling the flow of data, for example by enabling dynamic path selection based on constraints such as bandwidth availability, latency, and link costs. Its primary goals are to prevent network congestion, balance traffic loads, and ensure efficient use of bandwidth while meeting performance requirements.

The use cases for capacity planning, simulation, network simulation and network emulation, closed loop, and potentially others, should consider TE if configured in the network.

3.1.4. Closed Loop

A network closed loop refers to an automated and intelligent system where network operations are continuously monitored, analysed, and optimized in real time through feedback mechanisms. This self-adjusting cycle ensures that the network dynamically adapts to changes, resolves issues proactively, and maintains optimal performance without manual intervention.

Key Characteristics of a network closed loop:

* Real-time monitoring: Collects data from network devices, traffic flows, and applications to build a comprehensive view of network health and performance.

* Automated analysis: Identify anomalies, predict potential failures, or detect security threats, for example leveraging AI and machine learning.

* Proactive action: Automatically triggers corrective measures, such as reconfiguring devices, isolating compromised endpoints, or rerouting traffic.

* Continuous optimization: Uses feedback from previous cycles to refine network policies and improve future responses.

The SIMAP client application will be able to retrieve a topology layer and any network/node/termination point/link instances from the SIMAP server via the SIMAP APIs and from the response it will be able to map the traffic analysis to the entities (typically links and router) for automated analysis. The corrective measures would be applied, either directly to the network by managing the SIMAP entities (network/node/termination point/link instances) or by first validating the corrective measure in an offline simulation (see the simulation and traffic engineering use cases).

3.2. Inventory Queries

A network inventory refers to a comprehensive record or database that tracks and documents all the network components and devices within an organization’s IT infrastructure.

Key elements typically found in a network inventory include:

* Hardware details: Descriptions of physical devices such as routers (including their internal components such as cards, power supply units, pluggables), switches, servers, network cables, including model numbers, serial numbers, and manufacturer information. This information will facilitate locating additional details of the hardware in the manufacturer systems and the correlation with the purchase catalog of the company.

* Software and firmware: Versions of operating systems, network management tools, and firmware running on network devices. Note that a network device can have components with their own software and firmware.

* Licensing information: For any licensed software or devices, the network inventory will track license numbers, expiry dates, and compliance.

A network inventory lifecycle refers to the stages a network device or component goes through from its introduction to the network until its removal or replacement. It encompasses everything from acquisition and deployment to maintenance, upgrade, and eventually decommissioning. Managing the network inventory lifecycle efficiently is crucial for maintaining a secure, functional, and cost-effective network.

A well-maintained network inventory helps organizations with network management, troubleshooting, asset tracking, security, and ensuring compliance with regulations. It also helps in scaling the network, planning upgrades, and responding to issues quickly. In order to facilitate the planning and troubleshooting processes it is necessary to be able to navigate from network inventory to network topology and Services.

The SIMAP client application will be able to retrieve physical topology from the SIMAP server via the SIMAP APIs and from the response it will be able to retrieve the physical inventory of individual devices and cables and the customer information, if applicable.

The SIMAP client application may request either one or multiple topology layers via the SIMAP APIs and from the response it will be able to navigate to any other data modes outside of the core SIMAP topology to retrieve both physical and logical inventory.

For access network providers the ability to have linkage in the SIMAP of the complete network (active + passive) is essential as it provides many advantages for optimized customer Service, reduced Mean Time To Repair (MTTR), and lower operational costs through truck roll reduction. For example, operators may use custom-tags that are readily available for a customer-facing device, then query the inventory based on that tag to correlate it with the inventory and then map it to the network/service topology. The mapping and correlation can then be used for triggering appropriate Service checks.

The IVY working group is a good source of information for inventory information.

3.3. Service Placement Feasibility Checks

Service placement feasibility checks refer to the process of evaluating whether a specific Service can be deployed and operated effectively in a given network. This includes accessing the various factors to ensure that the service will function as intended (e.g., based on traffic performance requirements) without causing network disruptions or inefficiencies and effecting other Services already provisioned on the network.

Some of the factors that need assesing are network capabilities, status, limitations, resource usage and availability. The Service could be simulated during the feasibility checks to identify if there are any potential issues. The load testing could be done to evaluate performance under stress.

The service placement feasibility check application will be able to retrieve the topology at any layer from the SIMAP server via the SIMAP APIs and from the response it will be able to navigate to any other data models outside of the core SIMAP topology to retrieve any other information needed, such as resource usage, availability, status, etc.

3.4. Intent/Service Assurance

Network intent and Service assurance work together to ensure that the network aligns with business goals and that the Services provided meet the agreed-upon Service Level Agreements (SLAs).

The Service Assurance for Intent-Based Networking Architecture (SAIN) [RFC9417] approach emphasizes a comprehensive view of components involved in Service delivery, including network devices and functions, to effectively monitor and maintain Service health.

The key objectives of this architecture include:

* Holistic service monitoring: By considering all elements involved in Service delivery, the architecture enables a thorough assessment of service health.

* Correlation of Service degradation: It assists in linking Service performance issues to specific network components, facilitating precise identification of faults.

* Impact assessment: The architecture identifies which Services are affected by the failure or degradation of particular network components, aiding in prioritizing remediation efforts.

When a Service is degraded, the SAIN architecture will highlight where to look in the assurance Service graph, as opposed to going hop by hop to troubleshoot the issue. More precisely, the SAIN architecture will associate a list of symptoms originating from specific SAIN subservices to each Service instance, corresponding to components of the network. These components are good candidates for explaining the source of a Service degradation.

The SIMAP client application will be able to retrieve a topology layer and any network/node/termination point/link instances from the SIMAP server via the SIMAP APIs and from the response it will be able to determine the health of each instance by navigating to the SAIN subservices and its symptoms.

3.5. Service E2E and Per-link KPIs

The SIMAP client application will be able to retrieve a topology at any layer from a SIMAP server via the SIMAP APIs and from the response it will be able to navigate to and retrieve any KPIs for selected topology entity.

3.6. Network Capacity Planning

Network capacity planning refers to the process of analysing, predicting, and ensuring that the network has sufficient capacity (e.g., [RFC5136]), resources, and infrastructure to meet current and future demands. It involves evaluating the network's ability to handle increasing (including forecasted) amounts of data, traffic, and users' activity, while maintaining acceptable levels of performance, reliability, and security.

The capacity planning primary goal is to ensure that a network can support business operations, applications, and services without interruptions, delays, or degradation in quality. This requires a thorough understanding of the network's current state, as well as future requirements and growth projections.

Key aspects of network capacity planning include:

* Traffic analysis: Monitoring and analysing network traffic patterns to identify trends, peak usage periods, and areas of congestion. For example, by generating a core traffic matrix with IPFIX flow record [RFC7011] or deducting an approximate traffic matrix from the link utilization data.

* Resource utilization: Evaluating the link utilization throughout the network for the current demand to identify bottlenecks and potential QoS performance issues.

* Growth forecasting: Predicting future network growth based on business expansion, new applications, or changes in users' behavior.

* What-if scenarios: Creating models to assess the network behavior under different scenarios, such as increased traffic, failure conditions (link, router or Shared Risk Resource Group), and new application deployments (such as a new Content Delivery Network source, a new peering point, a new data center...).

* Upgrade planning: Identifying areas where upgrades or additions are needed to ensure that the network can minimize the effect of node/link failures, mitigate QoS problems, or simply to support growing demands.

* Cost-benefit analysis: Evaluating the costs and benefits of upgrading or adding new resources to determine the most cost-effective solutions.

By implementing a robust capacity planning process, organizations can:

* Ensure better network reliability: Minimize downtime and ensure that the network is always available when needed.

* Improve performance: Optimize network resources to support business-critical applications and Services.

* Optimize costs: Avoid unnecessary over-provisioning by making informed decisions based on data-driven insights.

* Support business growth: Scale the network to meet increasing demands and support business expansion.

The capacity planning application will be able to retrieve a topology layer and any network/node/termination point/link instances from the SIMAP server via the SIMAP APIs and from the response it will be able to map the traffic analysis to the entities (typically links and router), evaluate their current utilization, evaluate which elements to add to the network based on the growth forecasting, and finally perform the 'what-if' failure analysis by simulating the removal of link(s) and/or router(s) while evaluating the network performance.

3.7. Network Design

Network design involves defining both the logical structure, such as access, aggregation, and core layers, and the physical layout, including devices and links.

It serves as a blueprint, detailing how these elements interconnect to deliver the intended network behavior and functionality. The application will generate a candidate network topology, based on the initial design and the current network topology; this candidate network topology can then undergo further analysis (e.g., perform traffic flow simulations to identify bottlenecks and redundancy checks to ensure resilience) before being transformed into actionable intent and, eventually, deployment actions.

Throughout the network's lifecycle, the design rules embedded within a topology can be continuously validated. For example, a link rule might specify that a connection between core and aggregation layers must have its source(s) and destination(s) located within the same data center. Another example is to declare that a specific link type should only exist between Core <-> Aggregation layer with certain constraints on port optic speed, type (LR vs SR for instance), etc.

The network design application can (via SIMAP API):

* Write the intended network interconnect (topology + rules), this is the intent of the network topology that cannot be retrieved from the real network (e.g. our L2 topology interconnect intent, or L3 topology interconnect intent). One network (in case of small network) or interconnect of multiple networks (bigger networks).

* Retrieve the proposed network interconnect (topology + rules)

  - Use case can be for purpose of traffic simulation, testing behavior under failures. Network simulation use case is described in Section 3.8.

  - Use case can be for purpose of comparing different proposed network interconnects.

  - Use case can be to build a simulated environment using this design. Network simulation use case is described in Section 3.8.

* Retrieve the intended network interconnect (topology + rules)

* At any point in time, compare the discovered topology with intended one

  - Potentially validating discovered device configurations with intended ones assuming SIMAP has the external reference to configuration from topology.

3.8. Network Simulation and Network Emulation

Network simulation is a process used to analyse the behaviour of networks via software. It allows network engineers and researchers to assess how the network protocols work under different conditions, such as different topologies, traffic loads, network failures, or the introduction of new devices. Network emulation, on the other hand, replicates the behavior of a real-world network, allowing for more realistic analysis compared to network simulation. While network

simulation focuses on modeling and approximating network behavior, network emulation involves creating a real-time, functional network environment whose protocols behave exactly like a real network. Ideally, network emulation uses the same software images as the real network, but it could also be performed (with less accuracy) using generic software.

3.8.1. Types of Network Simulation

There are several types of network simulations, each designed to address specific needs and use cases. Below are the main categories of network simulation:

1. Discrete event simulation: This is the most common type of network simulation. It models a series of events that occur at specific points in time. Each event triggers a change in the state of a network component (e.g., a link is down, a card fails, or a packet arrives).

2. Continuous simulation: In contrast to discrete event simulation, continuous simulation models systems where variables change continuously over time. Network parameters like bandwidth, congestion, and throughput can be treated as continuous functions.

   The main use case is to model certain aspects of network performance that evolve continuously, such as link speeds or delay distributions in links that are impacted by environmental conditions (such as microwave or satellite links).

3. Monte Carlo simulation: This type of simulation uses statistical methods to model and analyse networks under uncertain or variable conditions. Monte Carlo simulations generate a large number of random samples to predict the performance of a network across multiple scenarios. It is used for probabilistic analysis, risk assessment, and performance evaluation under uncertain conditions.

3.8.2. Goals of Network Simulation

The simulations can be also classified depending on the goal of the simulation.

3.8.2.1. Network Protocol Analysis

This type of simulation focuses on simulating specific networking protocols (IS-IS, OSPF, BGP, SR) to understand how they perform under different conditions. It models the protocol operations and interactions amongst devices in the network. For example, simulation can be used to assess the impact of changing a link metric. Moreover, specific features of the networking protocol can be tested. For example, how fast-reroute performs in a given network topology.

3.8.2.2. Traffic Simulation

This simulation focuses on modelling traffic flow across the network, including packet generation, flow control, routing, and congestion. It aims to evaluate traffic's impact on network performance.

The main use is to model the impact of different types of traffic (e.g., voice, video, mobile data, web browsing) and understand how they affect the network's bandwidth and congestion levels. It can be used to identify bottlenecks and assist the capacity planning process.

3.8.2.3. Simulation of Different Topologies Under Normal and Failure Scenarios

This type of simulation focuses on the structure and layout of the network itself. It simulates different network topologies and their impact on the network's performance. It can be used, together with the traffic simulation, to evaluate the most efficient topology for a network under normal conditions and considering factors like fault tolerance.

3.9. Postmortem Replay

For the postmortem replay use case, the client application will use the SIMAP APIs for the purpose of analysis of network Service property evolution based on recorded changes. A collection of relevant timestamped network events, such as routing updates, configuration changes, link status modifications, traffic metrics evolution, and Service characteristics, is being made accessible from and/or within a SIMAP to support investigation and automated processing. Using a structured format, the stored data can be replayed in sequence, allowing network operators to examine historical network behavior, diagnose issues, and assess the impact of such events on Service assurance.

The mechanism supports correlation with external data sources to facilitate comprehensive post-mortem analysis. Beyond centralizing and correlating such various sources of information, the SIMAP can provide simulation of the network behaviour to assist investigations.

In essence, this use case builds upon a collection of other SIMAP use cases, such as inventory queries, intent/service assurance, Service KPIs, capacity planning, and simulation, to provide a thorough understanding of a network event impacting Service assurance.

Note that this use case may serve as a component of Service Disruption Detection fine-tuning as described in [I-D.ietf-nmop-network-anomaly-architecture].

3.10. Network Digital Twin (NDT)

Per [I-D.irtf-nmrg-network-digital-twin-arch], Network Digital Twin (NDT) is a digital representation that is used in the context of Networking and whose physical counterpart is a data network (e.g., provider network or enterprise network). Also, as discussed in Section 9.2 of [I-D.irtf-nmrg-network-digital-twin-arch], network element models and topology models help generate a virtual twin of the network according to the network element configuration, operation data, network topology relationship, link state and other network information. The operation status can be monitored and displayed, the network configuration change and optimization strategy can be pre-verified and historical data can support e.g. postmorem replay (Section 3.9))

Section 9.4 of [I-D.irtf-nmrg-network-digital-twin-arch] further elaborates on the requirements on various interfaces:

* Network-facing interfaces are twin interfaces between the real network and its twin entity. They are responsible for the information exchange between a real network and NDT. SIMAP APIs can be used within such interfaces.

* Application-facing interfaces are between the NDT and applications. They are responsible for the information exchange between Network Digital Twin and network applications. SIMAP APIs can be used for specification of hypothetical network and service states for ’what-if’ analysis, e.g.. for feasibility checks (Section 3.3), simulation or emulation (Section 3.8)). Such analysis may be used in support of e.g. network capacity planning (Section 3.6)) or network design (Section 3.7)).

Section 9.4 of [I-D.irtf-nmrg-network-digital-twin-arch] recommends that these interfaces are open and standardized so as to avoid either hardware or software vendor lock and achieve interoperability.

While network emulation (Section 3.8) can be a component within an NDT, the NDT itself is a broader construct that integrates multiple modeling techniques, including emulation, simulation, and analytics, to support intelligent network operations. NDT uses network emulation and includes network emulation use case, but it also interacts with the real network to support intelligent operations, including predictive analytics, intent verification, and full lifecycle management of network and services.

4. SIMAP Operator Requirements

The SIMAP operator requirements are split into three groups for different target audiences:

* Functional requirements: These requirements are collected from the operators and derived from the operators' use cases. Some of the more specific semantic requirements were identified as [RFC8345] gaps during the Hackathons with operators and added as specific semantic requirements to the operator use cases.

* Design requirements: These requirements are derived from the operator requirements. Although there is some duplication, these are focused on summarizing the operators' requirements for the design of the data model and API. These are functional requirements translated into low-level requirements for the model designers. The rationale for adopting this approach is to ensure that the data model is designed according to the operators' requirements and that they could be used for both design and review of the candidate data models.

* Architecture requirements: Architectural (non-functional) requirements are captured as well, as operators identified performance needs, large scale support, and network discovery. These are not data model requirements, but are requirements either to drive the APIs design itself (e.g., to better optimize performance) or for the SIMAP servers that expose a SIMAP API. Although, they may be common sense requirements not specific to SIMAP API, they are listed here for completeness.

4.1. Functional Requirements

The following are the operators' requirements for the SIMAP. Note that some of these requirements are supported by default by [RFC8345].

REQ-BASIC-MODEL-SUPPORT: Basic model with network, node, link, and termination point entity types.

This means that users of SIMAP must be able to reconstruct a topology at any layer in an unambiguous manner via these core concepts only, without the need to understand layer or technology specific information.

REQ-LAYERED-MODEL: Topology layers from physical layer up to Service layer.

SIMAP must provide views for all layers of network topology, from physical network (ideally optical), Layer 2, Layer 3 up to Service and intent views. It must provide flexibility to support both the same network topology instance with multiple layers (e.g., Layer 2 and Layer 3) or separate network topology instances with supporting relations between them (e.g., separate Layer 2 and Layer 3). Multiple topology layers can be grouped into the same network topology instance, if solution requires.

REQ-VIEWPOINTS: SIMAP should provide different views to different client applications. For example, one client application may need to see Layer 2 and Layer 3 layers in a single network topology instance, while another client application may need to see them as separate network topology instances.

REQ-PASSIVE-TOPO: SIMAP must support capability to model topology of the complete network. If the implementation requires passive topology to be part of the complete multi-layered topology, then SIMAP must support the capability to model the passive part of the network (in addition to the active part).

For access network providers the ability to have linkage in the SIMAP of the complete network (active + passive) is essential as it provides many advantages for optimized customer Service, reduced MTTR, and lower operational costs through truck roll reduction.

The passive topology must be either implemented in the SIMAP (what cannot be discovered can be added using the write API) or accessible from the SIMAP. Whether the passive topology is included as part of the SIMAP or accessible from the SIMAP is left to the solutions.

REQ-PROG-OPEN-MODEL: Open and programmable SIMAP.

This includes "read" operations to retrieve the view of the

network, typically as application-facing interface of Software Defined Networking (SDN) controllers or orchestrators.

It also includes "write" operations, not for the ability to directly change the live SIMAP data (e.g., changing the network or Service parameters), but for offline simulations, also known as what-if scenarios.

Running a "what-if" analysis requires the ability to take snapshots and to switch easily between them.

Note that there is a need to distinguish between a change on the SIMAP for future simulation and a change that reflects the current reality of the network.

SIMAP implementations and specifications MUST provide an unambiguous separation between real network topology state and simulation state. Simulation data MUST NOT overwrite, obscure, or be exposed as operational network state, and mechanisms MUST exist to ensure that simulated changes cannot be interpreted as real network conditions or configuration.

REQ-STD-API-BASED: Standard-based SIMAP and APIs, for multivendor support.

SIMAP must provide the standard APIs that provide for read/write and queries. These APIs must also provide the capability to retrieve the links to external data/models.

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

SIMAP and its APIs must be common over different network domains (campus, core, data center, etc.).

This means that clients of the SIMAP APIs must be able to understand the topology model of layers of any domain without having to understand the details of any technologies and domains.

REQ-GRAPH-TRAVERSAL: Topology graph traversal.

SIMAP must be optimized for graph traversal to support both network path queries and other specific use case queries. This means that the SIMAP must provide an efficient means to retrieve network paths, to accommodate the difficulty operators experience when retrieving network paths via the chain termination-point->link->termination-point, without having a direct adjacency relation. Additionally, SIMAP must enable efficient retrieval of the data required by other use case queries.

REQ-TOPOLOGY-ABSTRACTION: Navigation across the abstraction levels.

A network (even a single layer network) can be represented in multiple ways providing different abstraction views of the same network. In such a case, it would be beneficial being able to navigate amongst the different levels of abstractions (e.g. to understand which set of nodes in the native topology are actually represented as a single node in an abstract topology being built on top of the native topology). This navigation is different and orthogonal to the multi-layer navigation where we need to report which Layer 2 path is supporting a given Layer 3 node or link. Nevertheless, it would not be the best practice to expose it via different topology APIs and model. Please refer to the Appendix A.2 for some background on the topology abstraction.

SIMAP must provide a mechanism to navigate across the abstraction levels.

REQ-LIVE: Live network topology.

SIMAP must enable retrieval of multi-layered topology of a live network.

Live network is the latest snapshot of the real network.

REQ-SNAPSHOT: Network snapshot topology.

SIMAP must enable retrieval of multi-layered topology of different snapshots

Snapshot is the view of the network at any given point in time.

REQ-POTENTIAL: Potential new network topology.

SIMAP must enable both retrieval and write access to a potential network topology.

A potential new network topology is a provisional view at a given point in time that incorporates modifications relative to the current snapshot. It may represent the full topology or only the differences from the snapshot.

The view is needed for what-if analysis, pre-configuration, and simulation.

REQ-INTENDED: Intended network topology.

SIMAP must enable both retrieval and write access to the intended

network topology that cannot be discovered from the real network (e.g., intended Layer 2 Topology, intended Layer 3 Topology, and passive topology that cannot be discovered).

The intended topology represents the desired topology, without always detailing the intermediate hops, devices or detailed links that the current live topology contains. It can be used to verify if the live topology complies with the intent.

REQ-SEMANTIC: Network topology semantics.

SIMAP must provide semantics for layered network topologies and for linking external models/data.

The following requirements are more specific requirements for semantics:

REQ-LAYER-NAVIGATE: SIMAP must provide capability to navigate both within a topology layer and between topology layers.

Within-layer navigation means that SIMAP client applications should be able to move amongst entities that belong to the same layer. For example, in the IGP layer, the navigation should allow moving between OSPF/IS-IS management domains, OSPF/IS-IS areas, OSPF/IS-IS processes, OSPF/IS-IS interfaces, and OSPF/IS-IS links.

Between-layer navigation is the navigation across layers that should display the dependencies of entities in one layer on those in another. For instance, an IP interface that is supported by an Ethernet interface should be visible when moving between the corresponding layers.

REQ-EXTENSIBLE: SIMAP must be extensible with metadata. As examples, a controller or the client application could add a custom "location" attribute to a node to record its physical site, or a controller could attach a "vendorId" field to a device node. This demonstrates that arbitrary key-value metadata can be appended to any element in the model without altering the core schema.

REQ-PLUGG: SIMAP must be pluggable. That is,

* Must connect to other data models for device configuration, inventory, configuration, assurance, etc. The SIMAP does not contain the detailed device configuration, so a mechanism is needed to be able to link it from SIMAP. SIMAP should also be linked to a 'logical configuration inventory'. Several examples of the type of logical information to be linked from SIMAP: inventory of logical interfaces, inventory of ACLs, inventory of routing policies, or geographic location.

* Given that not all involved components can be available using YANG, there is a need to connect SIMAP with other modelling mechanisms.

REQ-BIDIR: SIMAP must provide a mechanism to model bidirectional links. 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., where a link is modeled as bidirectional, which in turn might be supported as unidirectional links at the lower layer.

REQ-MULTI-POINT: SIMAP must provide a mechanism to model multipoint links. A topology model should be able to model any topology type, including point to multipoint, bus, ring, star, tree, mesh, hybrid and daisy chain. A topology model should also be able to model any link cardinality, including point-to-point, point-to-multipoint, multipoint-to-multipoint

REQ-MULTI-DOMAIN: SIMAP must provide a mechanism to model links and nodes between networks when the implementation requires multi-domain topologies, topologies with multiple IGP areas or any network partitioning. This requirement is about covering connectivity between different networks, subnetworks, or domains.

REQ-SUBNETWORK: SIMAP must provide a mechanism to model network decomposition into subnetworks. The requirement is about modelling hierarchical networks , Autonomous Systems (ASes) with multiple areas, or a network with multiple domains (e.g., access, core, data center).

The network can be partitioned by providing capability to have multiple child network instances as part of a single parent network, with a relation between the parent network and child 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 a node or a node is supported by a network). This may be required to model supporting relationships for termination points which are supported by physical devices (e.g., a loopback interface on IP router).

REQ-STATUS: 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.

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

An SRv6 example:

In an SRv6-enabled network, the sourceIPv6Address field appears twice in the IPFIX data-template/data-record for a captured flow on an SRv6-enabled provider interface. Once in relation to provider data plane in the underlay, and once as relation to the customer data plane in the overlay.

SIMAP must provide the semantic capability that each sourceIPv6Address can be mapped to the overlay and underlay network topology. Both topologies might not be uniquely addressed, the VPN context (in SRv6 these are the SID’s, Section 3 of [RFC8986]) needs to be considered therefore as well.

IPFIX protocol, defined in [RFC7011], is the protocol for the exchange of flow information from an Exporting Process to a Collecting Process. Section 8 of [RFC7011] describes the management of Templates and Option templates at the Exporting and Collecting Processes, and states the following:

> | If an Information Element is required more than once in a Template, the different occurrences of this Information Element SHOULD follow the logical order of their treatments by the Metering Process. For example, if a selected packet goes through two hash functions, and if the two hash values are sent within a single Template, the first occurrence of the hash value should belong to the first hash function in the Metering Process. For example, when exporting the two source IP addresses of an IPv4-in-IPv4 packet, the first sourceIPv4Address Information Element occurrence should be the IPv4 address of the outer header, while the second occurrence should be the address of the inner header. Collecting Processes MUST properly handle Templates with multiple identical Information Elements.

REQ-CONTROL-PLANE: Control-plane routing state must be correlatable to the corresponding data-plane topology. For example, the underlay control-plane routing state must correlate to the underlay L3 topology, while the overlay control-plane routing state must correlate to the overlay L3 network topology.

A BMP/BGP example:

The BMP peer distinguisher (Section 4.2 of [RFC7854]) needs to be correlateable to the VRF of a node and the next-hop attribute of the NLRI in the BMP route-monitoring (Section 4.6 of [RFC7854]) encapsulated message to the underlay network topology while the path attribute of the NLRI in the BMP route-monitoring encapsulated message to the overlay topology.

4.2. Design Requirements

The following are the design requirements for the SIMAP data model:

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

SIMAP is not required to contain all models and data required for all the management and use cases. However, it should be designed to support adequate pointers to other functional data and models to ease navigating in the overall system. For example:

* ACLs and Route Policies are not required to be supported in the SIMAP, they would be linked to the SIMAP.

* Dynamic paths may, depending on the solution, be either inside or outside of the SIMAP. If outside of SIMAP, dynamic paths could be linked to the SIMAP.

SIMAP should ensure that it is possible to represent the paths/

routes and leave the choice of what level of dynamics to represent to the specific solution/implementations. The model needs to be rich enough to represent any level of dynamics. However, from experience, we suspect it can be the same model for all level of dynamics.

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

SIMAP entities should also provide information required to define semantics for layered network topologies, such as:

* link directionality,

* whether the links are multipoint or not and, if so, are whether these links are point-to-multipoint or multipoint-to-multipoint,

* role of the termination points in the link (source, destination, hub, spoke), and

* some generic mechanism to add metadata.

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

SIMAP should contain links to other models/data to enable generic navigation to other data models in generic way.

The SIMAP relationships should also provide information required to define semantics for layered network topologies, such as providing:

* underlay and overlay relations between different types of topological entities,

* additional information that helps with navigation inside a layer and between the layers, for example, easy identification of resources at the physical layer in primary versus backup paths, if the underlay resources are used for load balancing or for backup,

* capability to model nodes, termination points, and links contained in a network, but also nodes and links shared between networks, and

* relationships between networks, either for modelling of underlay and overlay or modelling network that contains multiple networks.

REQ-CONDITIONAL: Provide capability for conditional retrieval of parts of SIMAP.

REQ-TEMPO-HISTO: Must support geospatial (geographic coordinates, region, zone, etc.), temporal (when some fact is true, e.g., the topology or topological entity created at 12:00 UTC), and historical data (time-stamped historical changes, e.g. all changes from 2019-01-01 to 2023-06-30).

The geospatial, temporal and historical can also be supported external to the SIMAP.

4.3. Architectural Requirements

The following are the architectural requirements for the SIMAP server implementations that provide SIMAP API, they are the non-functional requirements for the SIMAP APIs and SIMAP server implementations:

REQ-SCALES: The SIMAP APIs and SIMAP server implementations must be scalable, it must support any provider network, independent of its size.

REQ-PERFORMANCE: The SIMAP APIs and SIMAP server implementations MUST support mechanisms that allow efficient retrieval of large topologies, including incremental, filtered, or paginated access to data. Implementations SHOULD support streaming or subscription-based mechanisms when appropriate to the protocol binding, to avoid requiring full-dataset retrieval for every request.

This requirement ensures that SIMAP can operate effectively in environments with large-scale, multi-layer topologies without mandating specific latency targets or performance metrics.

REQ-USABILITY: The SIMAP APIs must be simple and easy to integrate with the client applications, whose developers may not be networking experts.

REQ-DISCOVERY: A network SIMAP server must perform the initial and on-demand discovery of a network in order to provide the layered topology via the SIMAP APIs to a client application.

REQ-SYNCH: The SIMAP server must perform the sync with the network in order to provide up to date layered topology via SIMAP APIs to a client application

REQ-SECURITY: Any SIMAP interface MUST support strong client

authentication and authorization before granting access to SIMAP operations and data.

For YANG-based Netconf and RESTCONF protocols, access control SHOULD follow the Network Configuration Access Control Model (NACM) [RFC8341].

For non YANG protocols, implementations MUST provide an access control mechanism with similar level of protection to NACM, including fine grained authorization, role based access, and the ability to restrict access to sensitive topology and service and network data.

Because SIMAP connects highly sensitive multi layer topology with service and network data, implementations MUST ensure confidentiality, integrity, and replay protection for all SIMAP interactions, using security mechanisms appropriate to the transport protocol (e.g., TLS, SSH).

SIMAP implementations MUST prevent unauthorized write or simulation operations and ensure that simulation functions cannot become unintended configuration changes.

5. Positioning SIMAP in the Context of the IETF Work

[RFC8199] advocates for a consistent classification of YANG modules and introduces two abstraction layers for YANG modules:

* network element YANG modules

* network service YANG modules

The IRTF [RFC7426] defines the SDN layers and architecture and proposes the following interfaces:

* southbound interfaces between the network devices and controllers/ managers

* service interface between controllers/managers and applications

[RFC8309] defines where service model might fit into the SDN Architecture, although the service model does not require or preclude the use of SDN. It shows the following models at different layers of abstraction:

* device model, between network elements and controllers

* network model, between controllers and network orchestrators

* service model, between network orchestrators and service orchestrators

* customer service model, between service orchestrators and customer

[RFC8453] describes the ACTN architecture in the context of the YANG service models. It shows how ACTN interfaces relate to device model, network model and customer service model.

[RFC8969] describes a framework for Service and network management automation that takes advantage of YANG modelling technologies. This framework is drawn from a network operator perspective irrespective of the origin of a data model. [RFC8969] introduces "network service models" and describes the layering and representation of models within a network operator as follows:

* device model, between device and controller

* network model (operator oriented), between controller (that includes network orchestration function) and service orchestrator

* service model (customer oriented), between service orchestrator and customer, this is network service model

The SIMAP can be used at different layers of abstraction and SIMAP can provide topology at different interfaces. Although the SIMAP and APIs is primarily positioned as northbound multi-layered topology model from (SDN) Controllers, it can also be positioned as follows:

* In the context of [RFC8199], SIMAP can provide multi-layered topology YANG module as part of both network element and network service YANG modules

* In the context of [RFC7426], SIMAP can provide multi-layered topology interface as part of both Southbound and Service Interfaces

* In the context of [RFC8309], SIMAP can provide multi-layered topology model as part of device model, network model, service model and customer service model

* In the context of [RFC8453], SIMAP can provide multi-layered topology model as part of SBI (southbound interface to network), MPI (interface between multi-domain service coordinator and network controller) and CMI (interface between customer network controller and multi-domain service controller)

* In the context of [RFC8969], SIMAP can provide multi-layered topology model as part of device model, network model and network service model

Appendix A documents some other IETF activities related to topology modeling, it does not want to prescribe how SIMAP should be modeled or which base models should be used, and it is added for illustrational purposes only. Therefore it is not included in this section, but added to the Appendix.

6. Security Considerations

SIMAP provides a unified access point to multi layer topology, and relations to services and resources across network domains. Although this document defines concepts and implementation requirements rather than a concrete implementations and protocols, these requirements introduce significant security considerations that any SIMAP implementation or protocol MUST address.

Data sensitivity: SIMAP aggregates information that is highly sensitive in operational environments, including physical/logical/service topology, logical to physical relations, service relations, identifiers such as SRv6 SIDs, and references to configuration, inventory, assurance, and telemetry systems. Unauthorized read access to this information could enable targeted attacks, lateral movement, or large scale service disruption. Implementations MUST ensure that confidentiality protections and fine grained access control policies are applied to all SIMAP data.

Authentication: Any SIMAP implementation, regardless of modelling language or protocol, MUST provide strong client authentication before granting access to SIMAP data or operations. Authentication mechanisms depend on the underlying protocol binding (e.g., TLS client certificates, SSH keys, OAuth based API authentication), but the requirement for strong authentication is universal.

Authorization and protocol binding scope: Because SIMAP explicitly allows non YANG protocol bindings (REQ PLUGG), NACM applies only to YANG based bindings. Non YANG bindings MUST provide an access control mechanism that offers protections equivalent to NACM, including role based authorization, fine grained access restrictions, and the ability to limit access to sensitive topology and service mapping information.

Write and simulation operations: SIMAP supports operations that may

modify data or simulate modifications (e.g., what if analysis). Even when write operations are limited to simulation, implementations MUST ensure that such operations cannot become unintended configuration changes, and that simulation results cannot be used to infer privileged or hidden information.

Cross domain aggregation: SIMAP may expose information aggregated from multiple administrative domains. Implementations MUST ensure that access control policies are consistently enforced across domains and that cross domain data leakage does not occur. Policy mismatches or inconsistent authorization models across domains can create unintended disclosure paths.

Transport security: SIMAP implementations MUST ensure confidentiality, integrity, and replay protection for all protocol exchanges, regardless of the underlying protocol binding. Transport layer security mechanisms such as TLS [RFC8446] or SSH [RFC4253] MUST be used where applicable.

These considerations are not exhaustive; protocol specifications and implementations of SIMAP MUST define additional security mechanisms appropriate to their deployment environments.

Section 8 of [RFC8345] discusses further security consideration for YANG, NECONF, RESTCONF and specifically for ietf-network and ietf-network-topology modules.

7. IANA Considerations

This document has no actions for IANA.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

8.2. Informative References

[ETSI-ZSM-019]
ETSI, "Zero-Touch Network and Service Management (ZSM); ZSM Framework for NaaS", ETSI GR ZSM 019 V1.1.1, January 2026, <https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/019/01.01.01_60/gr_ZSM019v010101p.pdf>.

[I-D.ietf-ivy-network-inventory-topology]
Wu, B., Boucadair, M., Zhou, C., and Q. Wu, "A YANG Network Data Model for Inventory Topology Mapping", Work in Progress, Internet-Draft, draft-ietf-ivy-network-inventory-topology-07, 19 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ivy-network-inventory-topology-07>.

[I-D.ietf-ivy-network-inventory-yang]
Yu, C., Belotti, S., Bouquier, J., Peruzzini, F., and P. Bedard, "A Base YANG Data Model for Network Inventory", Work in Progress, Internet-Draft, draft-ietf-ivy-network-inventory-yang-18, 27 May 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-ivy-network-inventory-yang-18>.

[I-D.ietf-nmop-network-anomaly-architecture]
Graf, T., Du, W., Francois, P., and A. H. Feng, "A Framework for a Network Anomaly Detection Architecture", Work in Progress, Internet-Draft, draft-ietf-nmop-network-anomaly-architecture-07, 18 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-anomaly-architecture-07>.

[I-D.ietf-nmop-network-incident-yang]
Hu, T., Contreras, L. M., Wu, Q., Davis, N., and C. Feng, "A YANG Data Model for Network Incident Management", Work in Progress, Internet-Draft, draft-ietf-nmop-network-incident-yang-08, 13 February 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-network-incident-yang-08>.

[I-D.ietf-nmop-terminology]
Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>.

[I-D.ietf-opsawg-ntw-attachment-circuit]
Boucadair, M., Roberts, R., de Dios, O. G., Barguil, S., and B. Wu, "A Network YANG Data Model for Attachment Circuits", Work in Progress, Internet-Draft, draft-ietf-opsawg-ntw-attachment-circuit-16, 23 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-ntw-attachment-circuit-16>.

[I-D.ietf-opsawg-teas-attachment-circuit]
Boucadair, M., Roberts, R., de Dios, O. G., Barguil, S., and B. Wu, "YANG Data Models for Bearers and 'Attachment Circuits'-as-a-Service (ACaaS)", Work in Progress, Internet-Draft, draft-ietf-opsawg-teas-attachment-circuit-20, 23 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-teas-attachment-circuit-20>.

[I-D.ietf-teas-te-topo-and-tunnel-modeling]
Bryskin, I., Beeram, V. P., Saad, T., and X. Liu, "TE Topology and Tunnel Modeling for Transport Networks", Work in Progress, Internet-Draft, draft-ietf-teas-te-topo-and-tunnel-modeling-06, 12 July 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-teas-te-topo-and-tunnel-modeling-06>.

[I-D.irtf-nmrg-network-digital-twin-arch]
Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-12, 27 February 2026, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-12>.

[I-D.ogondio-nmop-isis-topology]
de Dios, O. G., Barguil, S., Lopez, V., Ceccarelli, D., and B. Claise, "A YANG Data Model for Intermediate System to intermediate System (IS-IS) Topology", Work in Progress, Internet-Draft, draft-ogondio-nmop-isis-topology-01, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ogondio-nmop-isis-topology-01>.

[I-D.ogondio-opsawg-ospf-topology]
de Dios, O. G., Barguil, S., and V. Lopez, "A YANG Data Model for Open Shortest Path First (OSPF) Topology", Work in Progress, Internet-Draft, draft-ogondio-opsawg-ospf-topology-01, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-ogondio-opsawg-ospf-topology-01>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/rfc/rfc3444>.

[RFC5136] Chimento, P. and J. Ishac, "Defining Network Capacity", RFC 5136, DOI 10.17487/RFC5136, February 2008, <https://www.rfc-editor.org/rfc/rfc5136>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/rfc/rfc7011>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/rfc/rfc7426>.

[RFC7854] Scudder, J., Ed., Fernando, R., and S. Stuart, "BGP Monitoring Protocol (BMP)", RFC 7854, DOI 10.17487/RFC7854, June 2016, <https://www.rfc-editor.org/rfc/rfc7854>.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/rfc/rfc7926>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <https://www.rfc-editor.org/rfc/rfc8199>.

[RFC8299] Wu, Q., Ed., Litkowski, S., Tomotaki, L., and K. Ogaki, "YANG Data Model for L3VPN Service Delivery", RFC 8299, DOI 10.17487/RFC8299, January 2018, <https://www.rfc-editor.org/rfc/rfc8299>.

[RFC8309] Wu, Q., Liu, W., and A. Farrel, "Service Models Explained", RFC 8309, DOI 10.17487/RFC8309, January 2018, <https://www.rfc-editor.org/rfc/rfc8309>.

[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>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/rfc/rfc8453>.

[RFC8466] Wen, B., Fioccola, G., Ed., Xie, C., and L. Jalil, "A YANG Data Model for Layer 2 Virtual Private Network (L2VPN) Service Delivery", RFC 8466, DOI 10.17487/RFC8466, October 2018, <https://www.rfc-editor.org/rfc/rfc8466>.

[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>.

[RFC8944] Dong, J., Wei, X., Wu, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer 2 Network Topologies", RFC 8944, DOI 10.17487/RFC8944, November 2020, <https://www.rfc-editor.org/rfc/rfc8944>.

[RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <https://www.rfc-editor.org/rfc/rfc8969>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/rfc/rfc8986>.

[RFC9179] Hopps, C., "A YANG Grouping for Geographic Locations", RFC 9179, DOI 10.17487/RFC9179, February 2022, <https://www.rfc-editor.org/rfc/rfc9179>.

[RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <https://www.rfc-editor.org/rfc/rfc9182>.

[RFC9291] Boucadair, M., Ed., Gonzalez de Dios, O., Ed., Barguil, S., and L. Munoz, "A YANG Network Data Model for Layer 2 VPNs", RFC 9291, DOI 10.17487/RFC9291, September 2022, <https://www.rfc-editor.org/rfc/rfc9291>.

[RFC9408] Boucadair, M., Ed., Gonzalez de Dios, O., Barguil, S., Wu, Q., and V. Lopez, "A YANG Network Data Model for Service Attachment Points (SAPs)", RFC 9408, DOI 10.17487/RFC9408, June 2023, <https://www.rfc-editor.org/rfc/rfc9408>.

[RFC9417] Claise, B., Quilbeuf, J., Lopez, D., Voyer, D., and T. Arumugam, "Service Assurance for Intent-Based Networking Architecture", RFC 9417, DOI 10.17487/RFC9417, July 2023, <https://www.rfc-editor.org/rfc/rfc9417>.

[RFC9418] Claise, B., Quilbeuf, J., Lucente, P., Fasano, P., and T. Arumugam, "A YANG Data Model for Service Assurance", RFC 9418, DOI 10.17487/RFC9418, July 2023, <https://www.rfc-editor.org/rfc/rfc9418>.

[RFC9522] Farrel, A., Ed., "Overview and Principles of Internet Traffic Engineering", RFC 9522, DOI 10.17487/RFC9522, January 2024, <https://www.rfc-editor.org/rfc/rfc9522>.

Appendix A. Related IETF Activities

> Note: The models cited in this section are provided for illustration purposes. It is out of scope to recomend which models will be used as base to build the SIMAP.

A.1. Network Topology

Interestingly, we could not find any network topology definition in IETF RFCs (not even in [RFC8345]) or Internet-Drafts. However, it is mentioned in multiple documents. As an example, in Overview and Principles of Internet Traffic Engineering [RFC9522], which mentions:

> To conduct performance studies and to support planning of existing and future networks, a routing analysis may be performed to determine the paths the routing protocols will choose for various traffic demands, and to ascertain the utilization of network resources as traffic is routed through the network. Routing analysis captures the selection of paths through the network, the assignment of traffic across multiple feasible routes, and the multiplexing of IP traffic over traffic trunks (if such constructs exist) and over the underlying network infrastructure. A model of network topology is necessary to perform routing analysis. A network topology model may be extracted from:
>
> * Network architecture documents
>
> * Network designs
>
> * Information contained in router configuration files
>
> * Routing databases such as the link state database of an interior gateway protocol (IGP)
>
> * Routing tables
>
> * Automated tools that discover and collate network topology information.
>
> Topology information may also be derived from servers that monitor network state, and from servers that perform provisioning functions.

Another example is [RFC8453] that defines native topology, abstract topology, black topology, and grey topology, but all in the context of actual topology and physical topology that are not specifically defined.

A.2. Topology Abstraction

Please refer to the following documents for some background on topology abstractions:

* [RFC7926] defines topology abstraction.

* Section 5 of [RFC8453] describes the topology abstraction methods and discusses topology abstraction factors, types, and their context in the ACTN architecture.

* Section 3.13 of [RFC8795] defines abstract TE topologies.

* Section 4.1 of [RFC8795] defines native TE topologies.

* Section 4.4 of [RFC8795] describes how to deal with multiple abstract TE topologies provided by the same provider.

* Section 1.3 of [I-D.ietf-teas-te-topo-and-tunnel-modeling] gives some background on topology abstraction.

A.3. Core SIMAP Components

The following specifications are relevant to the core functions provided by the SIMAP:

* IETF network model and network topology model [RFC8345]

* A YANG grouping for geographic location [RFC9179]

* IETF modules that augment [RFC8345] for different technologies:

  - A YANG data model for Traffic Engineering (TE) Topologies [RFC8795]

  - A YANG data model for Layer 2 network topologies [RFC8944]

  - A YANG data model for OSFP topology [I-D.ogondio-opsawg-ospf-topology]

  - A YANG data model for IS-IS topology [I-D.ogondio-nmop-isis-topology]

A.4. Additional SIMAP Components

The SIMAP may need to link to the following models, some are already augmenting [RFC8345]:

* Service Attachment Point (SAP) [RFC9408], augments 'ietf-network' data model [RFC8345] by adding the SAP.

* SAIN [RFC9417] [RFC9418]

* Network Inventory Model [I-D.ietf-ivy-network-inventory-yang] focuses on physical and virtual inventory. Logical inventory is currently outside of the scope. It does not augment [RFC8345].

* [I-D.ietf-ivy-network-inventory-topology] correlates the network inventory with the general topology via RFC8345 augmentations that reference inventory.

* KPIs: delay, jitter, loss

* Attachment Circuits (ACs) [I-D.ietf-opsawg-ntw-attachment-circuit] and [I-D.ietf-opsawg-teas-attachment-circuit]

* Configuration: The L2SM [RFC8466], L3SM [RFC8299], L2NM [RFC9291], and L3NM [RFC9182]

* Incident Management for Network Services [I-D.ietf-nmop-network-incident-yang]

Acknowledgments

Many thanks to Mohamed Boucadair and Reshad Rahman for their valuable contributions, reviews, and comments. Many thanks to Adrian Farrel for his SIMAP suggestion and helping to agree the terminology. Many thanks to Chongfeng Xie, Dan Voyer, Brad Peters, Diego Lopez, Ignacio Dominguez Martinez-Casanueva, Alex Huang Feng, Italo Busi, Wu Bo, Sherif Mostafa, Christopher Janz, Rob Evans, Danielle Ceccarelli, Sergio Belotti, Aihua Guo and many others for their contributions, suggestions and comments.

Many thanks to Nigel Davis for the valuable discussions and his confirmation of the modelling requirements.

Contributors

Ahmed Elhassany
Swisscom
Email: Ahmed.Elhassany@swisscom.com

Authors' Addresses

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

Benoit Claise
Everything OPS
Email: benoit@everything-ops.net

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

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