Internet Engineering Task Force (IETF)
Request for Comments: 8969
Category: Informational
ISSN: 2070-1721

Q. Wu, Ed.
Huawei
M. Boucadair, Ed.
Orange
D. Lopez
Telefonica I+D
C. Xie
China Telecom
L. Geng
China Mobile

January 2021

A Framework for Automating Service and Network Management with YANG

Abstract

Data models provide a programmatic approach to represent services and networks. Concretely, they can be used to derive configuration information for network and service components, and state information that will be monitored and tracked. Data models can be used during the service and network management life cycle (e.g., service instantiation, service provisioning, service optimization, service monitoring, service diagnosing, and service assurance). Data models are also instrumental in the automation of network management, and they can provide closed-loop control for adaptive and deterministic service creation, delivery, and maintenance.

This document describes a framework for service and network management automation that takes advantage of YANG modeling technologies. This framework is drawn from a network operator perspective irrespective of the origin of a data model; thus, it can accommodate YANG modules that are developed outside the IETF.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8969.

Table of Contents

1. Introduction

Service management systems usually comprise service activation/provision and service operation. Current service delivery procedures, from the processing of customer requirements and orders to service delivery and operation, typically assume the manipulation of data sequentially into multiple Operations Support System (OSS) or Business Support System (BSS) applications that may be managed by different departments within the service provider's organization (e.g., billing factory, design factory, network operation center). Many of these applications have been developed in house over the years and operate in a silo mode. As a result:

* The lack of standard data input/output (i.e., data model) raises many challenges in system integration and often results in manual configuration tasks.

* Service fulfillment systems might have a limited visibility on the network state and may therefore have a slow response to network changes.

Software-Defined Networking (SDN) becomes crucial to address these challenges. SDN techniques are meant to automate the overall service delivery procedures and typically rely upon standard data models. These models are used not only to reflect service providers' savoir faire, but also to dynamically instantiate and enforce a set of service-inferred policies that best accommodate what has been defined and possibly negotiated with the customer. [RFC7149] provides a first tentative attempt to rationalize that service provider's view on the SDN space by identifying concrete technical domains that need to be considered and for which solutions can be provided. These include:

* Techniques for the dynamic discovery of topology, devices, and capabilities, along with relevant information and data models that are meant to precisely document such topology, devices, and their capabilities.

* Techniques for exposing network services [RFC8309] and their characteristics.

* Techniques used by service-derived dynamic resource allocation and policy enforcement schemes, so that networks can be programmed accordingly.

* Dynamic feedback mechanisms that are meant to assess how efficiently a given policy (or a set thereof) is enforced from a service fulfillment and assurance perspective.

Models are key for each of the four technical items above. Service and network management automation is an important step to improve the agility of network operations. Models are also important to ease integrating multi-vendor solutions.

YANG module [RFC7950] developers have taken both top-down and bottom-up approaches to develop modules [RFC8199] and to establish a mapping between a network technology and customer requirements at the top or abstracting common constructs from various network technologies at the bottom. At the time of writing this document (2020), there are many YANG data models, including configuration and service models, that have been specified or are being specified by the IETF. They cover many of the networking protocols and techniques. However, how these models work together to configure a function, manage a set of devices involved in a service, or provide a service is something that is not currently documented either within the IETF or other Standards Development Organizations (SDOs).

Many of the YANG modules listed in this document are used to exchange data between NETCONF/RESTCONF clients and servers [RFC6241][RFC8040]. Nevertheless, YANG is a transport-independent data modeling language. It can thus be used independently of NETCONF/RESTCONF. For example, YANG can be used to define abstract data structures [RFC8791] that can be manipulated by other protocols (e.g., [DOTS-DDOS]).

This document describes an architectural framework for service and network management automation (Section 3) that takes advantage of YANG modeling technologies and investigates how YANG data models at different layers interact with each other (e.g., Service Mapping,

model composition) in the context of service delivery and fulfillment (Section 4). Concretely, the following benefits can be provided:

* Vendor-agnostic interfaces managing a service and the underlying network are allowed.

* Movement from deployment schemes where vendor-specific network managers are required to a scheme where the entities that are responsible for orchestrating and controlling services and network resources provided by multi-vendor devices are unified is allowed.

* Data inheritance and reusability among the various architecture layers thus promoting a network-wise provisioning instead of device-specific configuration is eased.

* Dynamically feeding a decision-making process (e.g., Controllers, Orchestrators) with notifications that will trigger appropriate actions, allowing that decision-making process to continuously adjust a network (and thus the involved resources) to deliver the service that conforms to the intended parameters (service objectives) is allowed.

This framework is drawn from a network operator perspective irrespective of the origin of a data model; it can also accommodate YANG modules that are developed outside the IETF. The document covers service models that are used by an operator to expose its services and capture service requirements from the customers (including other operators). Nevertheless, the document does not elaborate on the communication protocol(s) that makes use of these service models in order to request and deliver a service. Such considerations are out of scope.

The document identifies a list of use cases to exemplify the proposed approach (Section 5), but it does not claim nor aim to be exhaustive. Appendix A lists some examples to illustrate the layered YANG modules view.

2. Terminology and Abbreviations

2.1. Terminology

The following terms are defined in [RFC8309] and [RFC8199] and are not redefined here:

* Network Operator

* Customer

* Service

* Data Model

* Service Model

* Network Element Model

In addition, the document makes use of the following terms:

Network Model:
: Describes a network-level abstraction (or a subset of aspects of a network infrastructure), including devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices. This model corresponds to the network configuration model discussed in [RFC8309].

It can be used by a network operator to allocate resources (e.g., tunnel resource, topology resource) for the service or schedule resources to meet the service requirements defined in a service model.

Network Domain:
Refers to a network partitioning that is usually followed by network operators to delimit parts of their network. "access network" and "core network" are examples of network domains.

Device Model:
Refers to the Network Element YANG data model described in [RFC8199] or the device configuration model discussed in [RFC8309].

Device models are also used to refer to model a function embedded in a device (e.g., Network Address Translation (NAT) [RFC8512], Access Control Lists (ACLs) [RFC8519]).

Pipe:
Refers to a communication scope where only one-to-one (1:1) communications are allowed. The scope can be identified between ingress and egress nodes, two service sites, etc.

Hose:
Refers to a communication scope where one-to-many (1:N) communications are allowed (e.g., one site to multiple sites).

Funnel:
Refers to a communication scope where many-to-one (N:1) communications are allowed.

2.2. Abbreviations

The following abbreviations are used in the document:

ACL	Access Control List
AS	Autonomous System
AP	Access Point
CE	Customer Edge
DBE	Data Border Element
E2E	End-to-End
ECA	Event Condition Action
L2VPN	Layer 2 Virtual Private Network
L3VPN	Layer 3 Virtual Private Network
L3SM	L3VPN Service Model
L3NM	L3VPN Network Model
NAT	Network Address Translation
OAM	Operations, Administration, and Maintenance
OWD	One-Way Delay
PE	Provider Edge
PM	Performance Monitoring
QoS	Quality of Service
RD	Route Distinguisher
RT	Route Target
SBE	Session Border Element
SDN	Software-Defined Networking
SP	Service Provider
TE	Traffic Engineering
VN	Virtual Network
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding

3. Architectural Concepts and Goals

3.1. Data Models: Layering and Representation

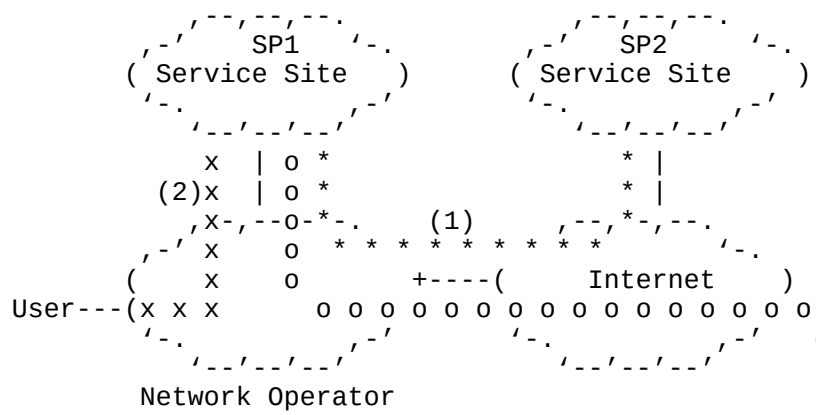
As described in Section 2 of [RFC8199], layering of modules allows for better reusability of lower-layer modules by higher-level modules while limiting duplication of features across layers.

Data models in the context of network management can be classified into service, network, and device models. Different service models may rely on the same set of network and/or device models.

Service models traditionally follow a top-down approach and are mostly customer-facing YANG modules providing a common model construct for higher-level network services (e.g., Layer 3 Virtual Private Network (L3VPN)). Such modules can be mapped to network technology-specific modules at lower layers (e.g., tunnel, routing, Quality of Service (QoS), security). For example, service models can be used to characterize the network service(s) to be ensured between service nodes (ingress/egress) such as:

* the communication scope (pipe, hose, funnel, etc.),
* the directionality (inbound/outbound),
* the traffic performance guarantees expressed using metrics such as One-Way Delay (OWD) [RFC7679] or One-Way Loss [RFC7680]; a summary of performance metrics maintained by IANA can be found in [IPPM],
* link capacity [RFC5136] [METRIC-METHOD],
* etc.

Figure 1 depicts the example of a Voice over IP (VoIP) service that relies upon connectivity services offered by a network operator. In this example, the VoIP service is offered to the network operator's customers by Service Provider 1 (SP1). In order to provide global VoIP reachability, SP1 Service Site interconnects with other Service Providers service sites typically by interconnecting Session Border Elements (SBEs) and Data Border Elements (DBEs) [RFC5486][RFC6406]. For other VoIP destinations, sessions are forwarded over the Internet. These connectivity services can be captured in a YANG service model that reflects the service attributes that are shown in Figure 2. This example follows the IP Connectivity Provisioning Profile template defined in [RFC7297].



**** (1) Inter-SP connectivity
xxxx (2) Customer-to-SP connectivity
oooo (3) SP to any destination connectivity

Figure 1: An Example of Service Connectivity Components

In reference to Figure 2, "Full traffic performance guarantees class" refers to a service class where all traffic performance metrics included in the service model (OWD, loss, delay variation) are guaranteed, while "Delay traffic performance guarantees class" refers to a service class where only OWD is guaranteed.

```
Connectivity: Scope and Guarantees
   (1) Inter-SP connectivity
      - Pipe scope from the local to the remote SBE/DBE
      - Full traffic performance guarantees class
   (2) Customer-to-SP connectivity
      - Hose/Funnel scope connecting the local SBE/DBE
        to the customer access points
      - Full traffic performance guarantees class
   (3) SP to any destination connectivity
      - Hose/Funnel scope from the local SBE/DBE to the
        Internet gateway
      - Delay traffic performance guarantees class
Flow Identification
   * Destination IP address (SBE, DBE)
   * DSCP marking
Traffic Isolation
   * VPN
Routing & Forwarding
   * Routing rule to exclude some ASes from the inter-domain
     paths
Notifications (including feedback)
   * Statistics on aggregate traffic to adjust capacity
   * Failures
   * Planned maintenance operations
   * Triggered by thresholds
```

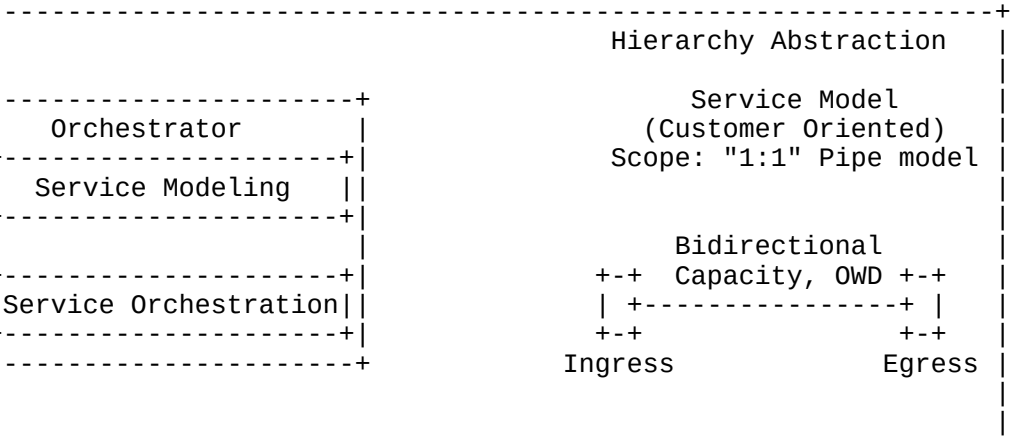
Figure 2: Sample Attributes Captured in a Service Model

Network models are mainly network-resource-facing modules; they describe various aspects of a network infrastructure, including devices and their subsystems, and relevant protocols operating at the link and network layers across multiple devices (e.g., network topology and traffic-engineering tunnel modules).

Device (and function) models usually follow a bottom-up approach and are mostly technology-specific modules used to realize a service (e.g., BGP, ACL).

Each level maintains a view of the supported YANG modules provided by lower levels (see for example, Appendix A). Mechanisms such as the YANG library [RFC8525] can be used to expose which YANG modules are supported by nodes in lower levels.

Figure 3 illustrates the overall layering model. The reader may refer to Section 4 of [RFC8309] for an overview of "Orchestrator" and "Controller" elements. All these elements (i.e., Orchestrator(s), Controller(s), device(s)) are under the responsibility of the same operator.



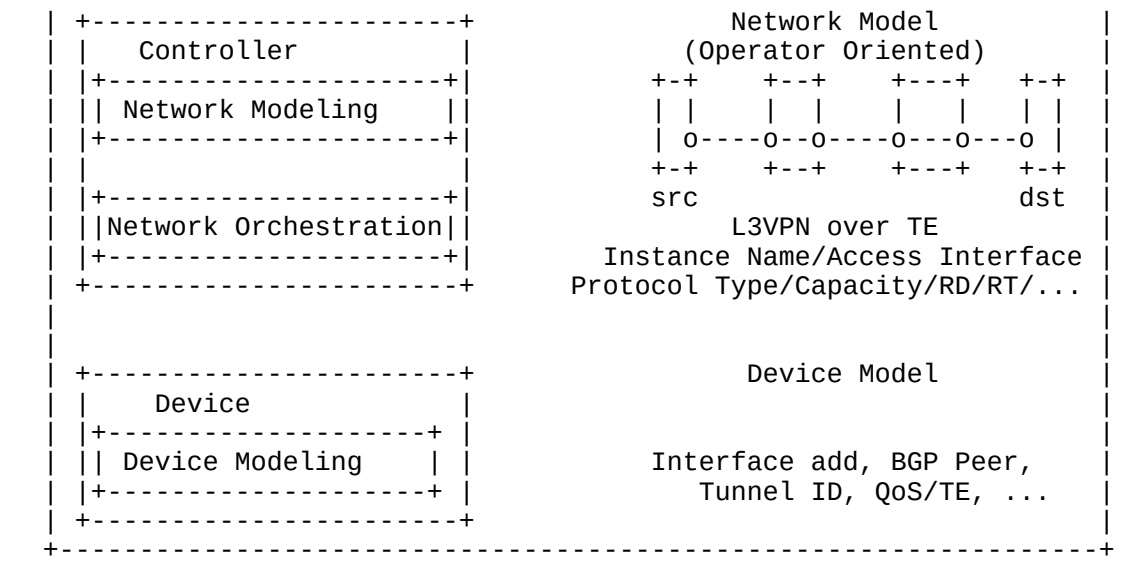


Figure 3: Layering and Representation within a Network Operator

A composite service offered by a network operator may rely on services from other operators. In such a case, the network operator acts as a customer to request services from other networks. The operators providing these services will then follow the layering depicted in Figure 3. The mapping between a composite service and a third-party service is maintained at the orchestration level. From a data-plane perspective, appropriate traffic steering policies (e.g., Service Function Chaining [RFC7665]) are managed by the network controllers to guide how/when a third-party service is invoked for flows bound to a composite service.

The layering model depicted in Figure 3 does not make any assumption about the location of the various entities (e.g., Controller, Orchestrator) within the network. As such, the architecture does not preclude deployments where, for example, the Controller is embedded on a device that hosts other functions that are controlled via YANG modules.

In order to ease the mapping between layers and data reuse, this document focuses on service models that are modeled using YANG. Nevertheless, fully compliant with Section 3 of [RFC8309], Figure 3 does not preclude service models to be modeled using data modeling languages other than YANG.

3.2. Automation of Service Delivery Procedures

Service models can be used by a network operator to expose its services to its customers. Exposing such models allows automation of the activation of service orders and thus the service delivery. One or more monolithic service models can be used in the context of a composite service activation request (e.g., delivery of a caching infrastructure over a VPN). Such models are used to feed a decision-making intelligence to adequately accommodate customer needs.

Also, such models may be used jointly with services that require dynamic invocation. An example is provided by the service modules defined by the DOTS WG to dynamically trigger requests to handle Distributed Denial-of-Service (DDoS) attacks [RFC8783]. The service filtering request modeled using [RFC8783] will be translated into device-specific filtering (e.g., ACLs defined in [RFC8519]) that fulfills the service request.

Network models can be derived from service models and used to

provision, monitor, and instantiate the service. Also, they are used to provide life-cycle management of network resources. Doing so is meant to:

* expose network resources to customers (including other network operators) to provide service fulfillment and assurance.

* allow customers (or network operators) to dynamically adjust the network resources based on service requirements as described in service models (e.g., Figure 2) and the current network performance information described in the telemetry modules.

Note that it is out of the scope of this document to elaborate on the communication protocols that are used to implement the interface between the service ordering (customer) and service order handling (provider).

3.3. Service Fulfillment Automation

To operate a service, the settings of the parameters in the device models are derived from service models and/or network models and are used to:

* Provision each involved network function/device with the proper configuration information.

* Operate the network based on service requirements as described in the service model(s) and local operational guidelines.

In addition, the operational state including configuration that is in effect together with statistics should be exposed to upper layers to provide better network visibility and assess to what extent the derived low-level modules are consistent with the upper-level inputs.

Filters are enforced on the notifications that are communicated to Service layers. The type and frequency of notifications may be agreed upon in the service model.

Note that it is important to correlate telemetry data with configuration data to be used for closed loops at the different stages of service delivery, from resource allocation to service operation, in particular.

3.4. YANG Module Integration

To support top-down service delivery, YANG modules at different levels or at the same level need to be integrated for proper service delivery (including proper network setup). For example, the service parameters captured in service models need to be decomposed into a set of configuration/notification parameters that may be specific to one or more technologies; these technology-specific parameters are grouped together to define technology-specific device-level models or network-level models.

In addition, these technology-specific device or network models can be further integrated with each other using the schema mount mechanism [RFC8528] to provision each involved network function/ device or each involved network domain to support newly added modules or features. A collection of integrated device models can be loaded and validated during implementation.

High-level policies can be defined at service or network models (e.g., "Autonomous System Number (ASN) Exclude" in the example depicted in Figure 2). Device models will be tweaked accordingly to provide policy-based management. Policies can also be used for telemetry automation, e.g., policies that contain conditions to

trigger the generation and pushing of new telemetry data.

4. Functional Blocks and Interactions

The architectural considerations described in Section 3 lead to the life-cycle management architecture illustrated in Figure 4 and described in the following subsections.

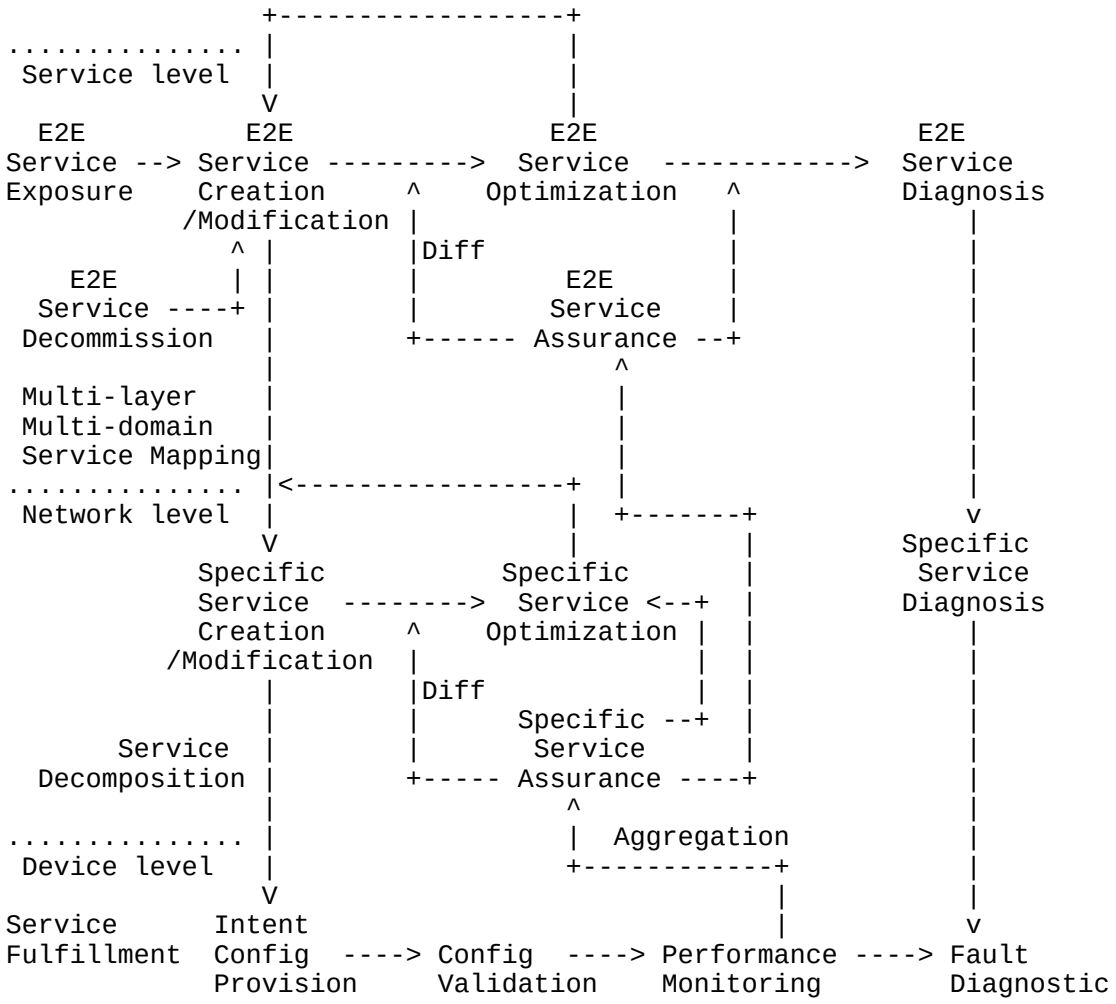


Figure 4: Service and Network Life-Cycle Management

4.1. Service Life-Cycle Management Procedure

Service life-cycle management includes end-to-end service life-cycle management at the service level and technology-specific network life-cycle management at the network level.

The end-to-end service life-cycle management is technology-independent service management and spans across multiple network domains and/or multiple layers while technology-specific service life-cycle management is technology domain-specific or layer-specific service life-cycle management.

4.1.1. Service Exposure

A service in the context of this document (sometimes called "Network Service") is some form of connectivity between customer sites and the Internet or between customer sites across the operator’s network and across the Internet.

Service exposure is used to capture services offered to customers (ordering and order handling). One example is that a customer can

use an L3VPN Service Model (L3SM) to request L3VPN service by providing the abstract technical characterization of the intended service between customer sites.

Service model catalogs can be created to expose the various services and the information needed to invoke/order a given service.

4.1.2. Service Creation/Modification

A customer is usually unaware of the technology that the network operator has available to deliver the service, so the customer does not make requests specific to the underlying technology but is limited to making requests specific to the service that is to be delivered. This service request can be filled using a service model.

Upon receiving a service request, and assuming that appropriate authentication and authorization checks have been made with success, the service Orchestrator/management system should verify whether the service requirements in the service request can be met (i.e., whether there are sufficient resources that can be allocated with the requested guarantees).

If the request is accepted, the service Orchestrator/management system maps such a service request to its view. This view can be described as a technology-specific network model or a set of technology-specific device models, and this mapping may include a choice of which networks and technologies to use depending on which service features have been requested.

In addition, a customer may require a change in the underlying network infrastructure to adapt to new customers' needs and service requirements (e.g., service a new customer site, add a new access link, or provide disjoint paths). This service modification can be issued following the same service model used by the service request.

Withdrawing a service is discussed in Section 4.1.6.

4.1.3. Service Assurance

The performance measurement telemetry (Section 4.2.3) can be used to provide service assurance at service and/or network levels. The performance measurement telemetry model can tie with service or network models to monitor network performance or Service Level Agreements.

4.1.4. Service Optimization

Service optimization is a technique that gets the configuration of the network updated due to network changes, incident mitigation, or new service requirements. One example is once a tunnel or a VPN is set up, performance monitoring information or telemetry information per tunnel (or per VPN) can be collected and fed into the management system. If the network performance doesn't meet the service requirements, the management system can create new VPN policies capturing network service requirements and populate them into the network.

Both network performance information and policies can be modeled using YANG. With Policy-based management, self-configuration and self-optimization behavior can be specified and implemented.

The overall service optimization is managed at the service level, while the network level is responsible for the optimization of the specific network services it provides.

4.1.5. Service Diagnosis

Operations, Administration, and Maintenance (OAM) are important networking functions for service diagnosis that allow network operators to:

* monitor network communications (i.e., reachability verification and Continuity Check)

* troubleshoot failures (i.e., fault verification and localization)

* monitor service level agreements and performance (i.e., performance management)

When the network is down, service diagnosis should be in place to pinpoint the problem and provide recommendations (or instructions) for network recovery.

The service diagnosis information can be modeled as technology-independent Remote Procedure Call (RPC) operations for OAM protocols and technology-independent abstraction of key OAM constructs for OAM protocols [RFC8531][RFC8533]. These models can be used to provide consistent configuration, reporting, and presentation for the OAM mechanisms used to manage the network.

Refer to Section 4.2.4 for the device-specific side.

4.1.6. Service Decommission

Service decommission allows a customer to stop the service by removing the service from active status, thus releasing the network resources that were allocated to the service. Customers can also use the service model to withdraw the subscription to a service.

4.2. Service Fulfillment Management Procedure

4.2.1. Intended Configuration Provision

Intended configuration at the device level is derived from network models at the network level or service models at the service level and represents the configuration that the system attempts to apply. Take L3SM as a service model example to deliver an L3VPN service; there is a need to map the L3VPN service view defined in the service model into a detailed intended configuration view defined by specific configuration models for network elements. The configuration information includes:

* Virtual Routing and Forwarding (VRF) definition, including VPN policy expression

* Physical Interface(s)

* IP layer (IPv4, IPv6)

* QoS features such as classification, profiles, etc.

* Routing protocols: support of configuration of all protocols listed in a service request, as well as routing policies associated with those protocols

* Multicast support

* Address sharing

* Security (e.g., access control, authentication, encryption)

These specific configuration models can be used to configure Provider

Edge (PE) and Customer Edge (CE) devices within a site, e.g., a BGP policy model can be used to establish VPN membership between sites and VPN service topology.

Note that in networks with legacy devices (that support proprietary modules or do not support YANG at all), an adaptation layer is likely to be required at the network level so that these devices can be involved in the delivery of the network services.

This interface is also used to handle service withdrawal (Section 4.1.6).

4.2.2. Configuration Validation

Configuration validation is used to validate intended configuration and ensure the configuration takes effect.

For example, if a customer creates an interface "eth-0/0/0" but the interface does not physically exist at this point, then configuration data appears in the <intended> status but does not appear in the <operational> datastore. More details about <intended> and <operational> datastores can be found in Section 5.1 of [RFC8342].

4.2.3. Performance Monitoring

When a configuration is in effect in a device, the <operational> datastore holds the complete operational state of the device, including learned, system, default configuration, and system state. However, the configurations and state of a particular device do not have visibility on the whole network, nor can they show how packets are going to be forwarded through the entire network. Therefore, it becomes more difficult to operate the entire network without understanding the current status of the network.

The management system should subscribe to updates of a YANG datastore in all the network devices for performance monitoring purposes and build a full topological visibility of the network by aggregating (and filtering) these operational states from different sources.

4.2.4. Fault Diagnostic

When configuration is in effect in a device, some devices may be misconfigured (e.g., device links are not consistent in both sides of the network connection) or network resources might be misallocated. Therefore, services may be negatively affected without knowing the root cause in the network.

Technology-dependent nodes and RPC commands are defined in technology-specific YANG data models, which can use and extend the base model described in Section 4.1.5 to deal with these issues.

These RPC commands received in the technology-dependent node can be used to trigger technology-specific OAM message exchanges for fault verification and fault isolation. For example, Transparent Interconnection of Lots of Links (TRILL) Multi-destination Tree Verification (MTV) RPC command [TRILL-YANG-OAM] can be used to trigger Multi-Destination Tree Verification Messages (MTVMs) defined in [RFC7455] to verify TRILL distribution tree integrity.

4.3. Multi-layer/Multi-domain Service Mapping

Multi-layer/Multi-domain Service Mapping allows the mapping of an end-to-end abstract view of the service segmented at different layers and/or different network domains into domain-specific views.

One example is to map service parameters in the L3SM into

configuration parameters such as Route Distinguisher (RD), Route Target (RT), and VRF in the L3VPN Network Model (L3NM).

Another example is to map service parameters in the L3SM into Traffic Engineered (TE) tunnel parameters (e.g., Tunnel ID) in TE model and Virtual Network (VN) parameters (e.g., Access Point (AP) list and VN members) in the YANG data model for VN operation [ACTN-VN-YANG].

4.4. Service Decomposition

Service Decomposition allows to decompose service models at the service level or network models at the network level into a set of device models at the device level. These device models may be tied to specific device types or classified into a collection of related YANG modules based on service types and features offered, and they may load at the implementation time before configuration is loaded and validated.

5. YANG Data Model Integration Examples

The following subsections provide some YANG data model integration examples.

5.1. L2VPN/L3VPN Service Delivery

In reference to Figure 5, the following steps are performed to deliver the L3VPN service within the network management automation architecture defined in Section 4:

1. The Customer requests to create two sites (as per Service Creation in Section 4.1.2) relying upon L3SM with each site having one network access connectivity, for example:

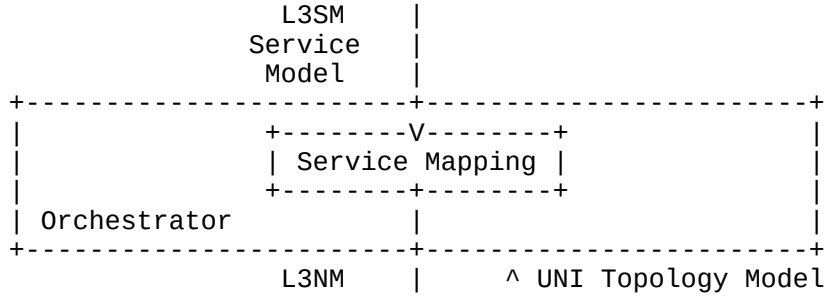
   * Site A: network-access A, link-capacity = 20 Mbps, class "foo", guaranteed-capacity-percent = 10, average-one-way-delay = 70 ms.

   * Site B: network-access B, link-capacity = 30 Mbps, class "foo1", guaranteed-capacity-percent = 15, average-one-way-delay = 60 ms.

2. The Orchestrator extracts the service parameters from the L3SM. Then, it uses them as input to the Service Mapping in Section 4.3 to translate them into orchestrated configuration parameters (e.g., RD, RT, and VRF) that are part of the L3NM specified in [OPSAWG-L3SM-L3NM].

3. The Controller takes the orchestrated configuration parameters in the L3NM and translates them into an orchestrated (Service Decomposition in Section 4.4) configuration of network elements that are part of, e.g., BGP, QoS, Network Instance, IP management, and interface models.

[UNI-TOPOLOGY] can be used for representing, managing, and controlling the User Network Interface (UNI) topology.



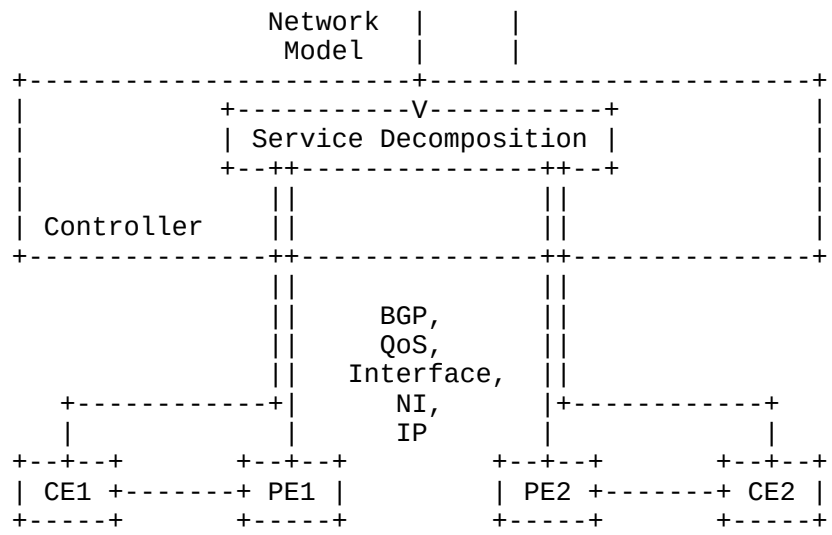


Figure 5: L3VPN Service Delivery Example (Current)

L3NM inherits some of the data elements from the L3SM. Nevertheless, the L3NM as designed in [OPSAWG-L3SM-L3NM] does not expose some information to the above layer such as the capabilities of an underlying network (which can be used to drive service order handling) or notifications (to notify subscribers about specific events or degradations as per agreed SLAs). Some of this information can be provided using, e.g., [OPSAWG-YANG-VPN]. A target overall model is depicted in Figure 6.

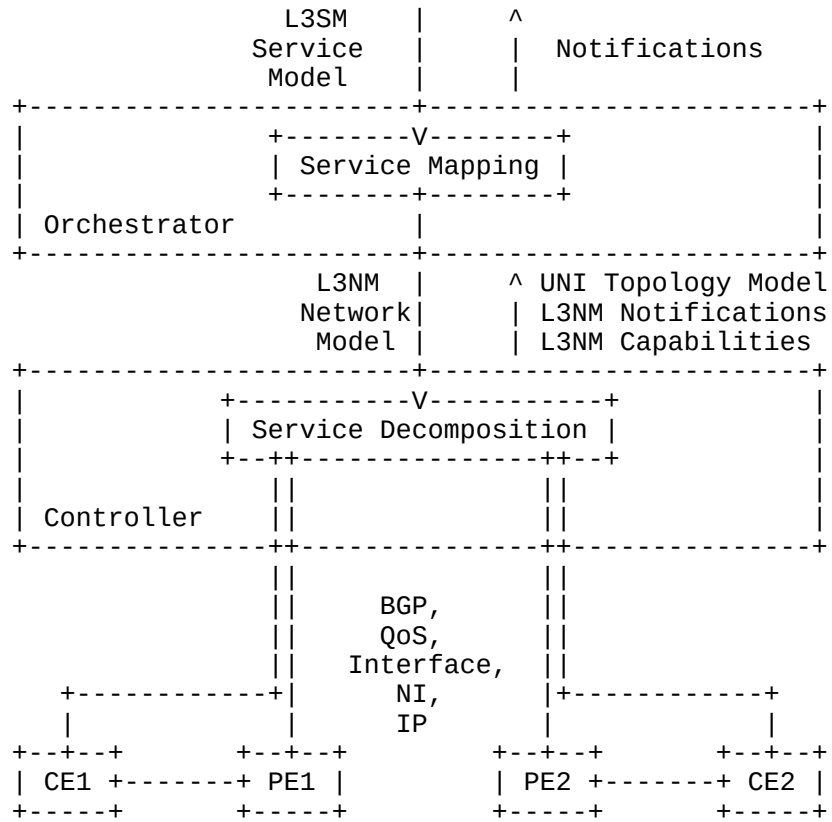


Figure 6: L3VPN Service Delivery Example (Target)

Note that a similar analysis can be performed for Layer 2 VPNs (L2VPNs). An L2VPN Service Model (L2SM) is defined in [RFC8466], while the YANG L2VPN Network Model (L2NM) is specified in [OPSAWG-L2NM].

5.2. VN Life-Cycle Management

In reference to Figure 7, the following steps are performed to deliver the VN service within the network management automation architecture defined in Section 4:

1. A customer makes a request (Service Exposure in Section 4.1.1) to create a VN. The association between the VN, APs, and VN members is defined in the VN YANG model [ACTN-VN-YANG].

2. The Orchestrator creates the single abstract node topology based on the information captured in the request.

3. The customer exchanges with the Orchestrator the connectivity matrix on the abstract node topology and explicit paths using the TE topology model [RFC8795]. This information can be used to instantiate the VN and set up tunnels between source and destination endpoints (Service Creation in Section 4.1.2).

4. In order to provide service assurance (Service Optimization in Section 4.1.4), the telemetry model that augments the VN model and corresponding TE tunnel model can be used by the Orchestrator to subscribe to performance measurement data. The Controller will then notify the Orchestrator with all the parameter changes and network performance changes related to the VN topology and the tunnels [TEAS-ACTN-PM].

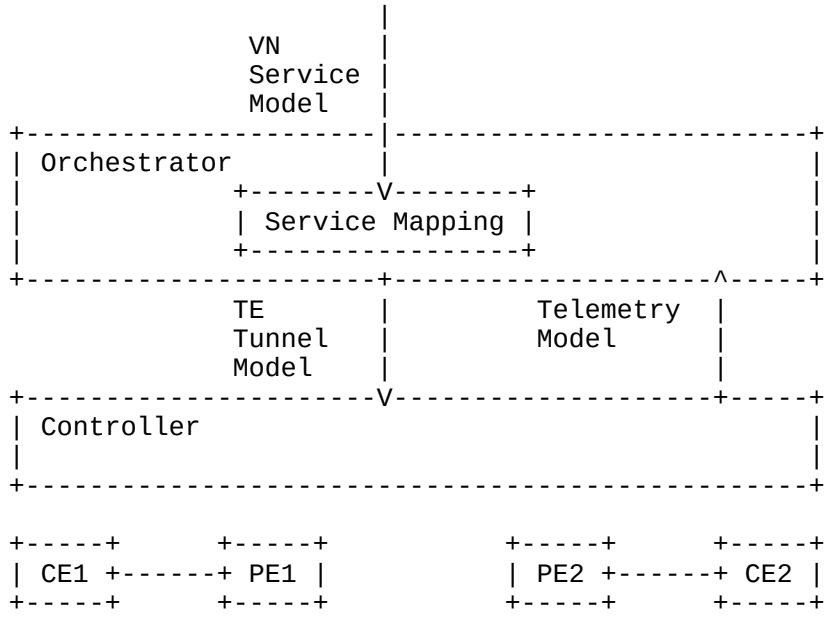


Figure 7: A VN Service Delivery Example

5.3. Event-Based Telemetry in the Device Self Management

In reference to Figure 8, the following steps are performed to monitor state changes of managed resources in a network device and provide device self management within the network management automation architecture defined in Section 4:

1. To control which state a network device should be in or is allowed to be in at any given time, a set of conditions and actions are defined and correlated with network events (e.g., allow the NETCONF server to send updates only when the value exceeds a certain threshold for the first time, but not again until the threshold is cleared), which constitute an Event Condition Action (ECA) policy or an event-driven policy control logic that can be executed on the device (e.g., [EVENT-YANG]).

2. To provide a rapid autonomic response that can exhibit self-management properties, the Controller pushes the ECA policy to the network device and delegates the network control logic to the network device.

3. The network device uses the ECA model to subscribe to the event source, e.g., an event stream or datastore state data conveyed to the server via YANG-Push subscription [RFC8641], monitors state parameters, and takes simple and instant actions when an associated event condition on state parameters is met. ECA notifications can be generated as the result of actions based on event stream subscription or datastore subscription (model-driven telemetry operation discussed in Section 4.2.3).

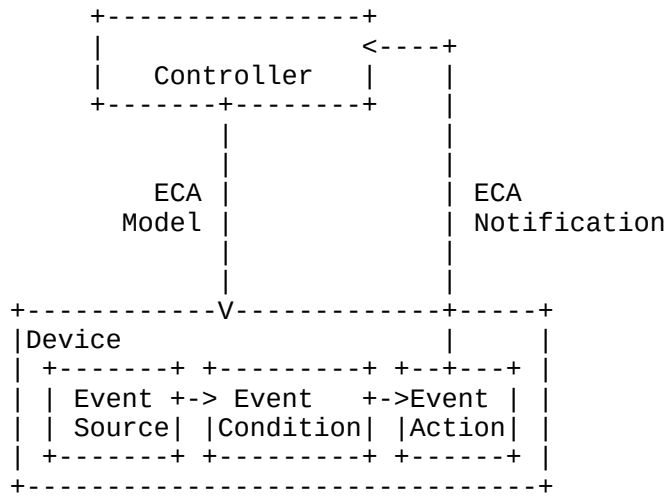


Figure 8: Event-Based Telemetry

6. Security Considerations

Many of the YANG modules cited in this document define schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

Security considerations specific to each of the technologies and protocols listed in the document are discussed in the specification documents of each of these protocols.

In order to prevent leaking sensitive information and the "confused deputy" problem [Hardy] in general, special care should be considered when translating between the various layers in Section 4 or when aggregating data retrieved from various sources. Authorization and authentication checks should be performed to ensure that data is available to an authorized entity. The network operator must enforce means to protect privacy-related information included in customer-facing models.

To detect misalignment between layers that might be induced by misbehaving nodes, upper layers should continuously monitor the perceived service (Section 4.1.4) and should proceed with checks to

assess that the provided service complies with the expected service and that the data reported by an underlying layer is matching the perceived service by the above layer. Such checks are the responsibility of the service diagnosis (Section 4.1.5).

When a YANG module includes security-related parameters, it is recommended to include the relevant information as part of the service assurance to track the correct functioning of the security mechanisms.

Additional considerations are discussed in the following subsections.

6.1. Service Level

A provider may rely on services offered by other providers to build composite services. Appropriate mechanisms should be enabled by the provider to monitor and detect a service disruption from these providers. The characterization of a service disruption (including mean time between failures and mean time to repair), the escalation procedure, and penalties are usually documented in contractual agreements (e.g., as described in Section 2.1 of [RFC4176]). Misbehaving peer providers will thus be identified and appropriate countermeasures will be applied.

The communication protocols that make use of a service model between a customer and an operator are out of scope. Relevant security considerations should be discussed in the specification documents of these protocols.

6.2. Network Level

Security considerations specific to the network level are listed below:

* A controller may create forwarding loops by misconfiguring the underlying network nodes. It is recommended to proceed with tests to check the status of forwarding paths regularly or whenever changes are made to routing or forwarding processes. Such checks may be triggered from the service level owing to the means discussed in Section 4.1.5.

* Some service models may include a traffic isolation clause that is passed down to the network level so that appropriate technology-specific actions must be enforced at the underlying network (and thus involved network devices) to avoid that such traffic is accessible to non-authorized parties. In particular, network models may indicate whether encryption is enabled and, if so, expose a list of supported encryption schemes and parameters. Refer, for example, to the encryption feature defined in [OPSAWG-VPN-COMMON] and its use in [OPSAWG-L3SM-L3NM].

6.3. Device Level

Network operators should monitor and audit their networks to detect misbehaving nodes and abnormal behaviors. For example, OAM, as discussed in Section 4.1.5, can be used for that purpose.

Access to some data requires specific access privilege levels. Devices must check that a required access privilege is provided before granting access to specific data or performing specific actions.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[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/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

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

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

[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/info/rfc8341>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Informative References

[ACTN-VN-YANG] Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Y. Yoon, "A YANG Data Model for VN Operation", Work in Progress, Internet-Draft, draft-ietf-teas-actn-vn-yang-10, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-teas-actn-vn-yang-10>.

[BFD-YANG] Rahman, R., Zheng, L., Jethanandani, M., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", Work in Progress, Internet-Draft, draft-ietf-bfd-yang-17, 2 August 2018, <https://tools.ietf.org/html/draft-ietf-bfd-yang-17>.

[DOTS-DDOS] Boucadair, M., Shallow, J., and T. Reddy.K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", Work in Progress, Internet-Draft, draft-ietf-dots-rfc8782-bis-04, 3 December 2020, <https://tools.ietf.org/html/draft-ietf-dots-rfc8782-bis-04>.

[EVENT-YANG] Wu, Q., Bryskin, I., Birkholz, H., Liu, X., and B. Claise, "A YANG Data model for ECA Policy Management", Work in Progress, Internet-Draft, draft-wwx-netmod-event-yang-10, 1 November 2020, <https://tools.ietf.org/html/draft-wwx-netmod-event-yang-10>.

[EVPN-YANG] Brissette, P., Shah, H., Hussain, I., Tiruveedhula, K., and J. Rabadan, "Yang Data Model for EVPN", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-yang-07, 11 March 2019, <https://tools.ietf.org/html/draft-ietf-bess-evpn-yang-07>.

[Hardy] Hardy, N., "The Confused Deputy: (or why capabilities might have been invented)", DOI 10.1145/54289.871709, October 1988, <https://dl.acm.org/doi/10.1145/54289.871709>.

[IDR-BGP-MODEL] Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-10, 15 November 2020, <https://tools.ietf.org/html/draft-ietf-idr-bgp-model-10>.

[IPPM] IANA, "Performance Metrics", March 2020, <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>.

[L2VPN-YANG] Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", Work in Progress, Internet-Draft, draft-ietf-bess-l2vpn-yang-10, 2 July 2019, <https://tools.ietf.org/html/draft-ietf-bess-l2vpn-yang-10>.

[L3VPN-YANG] Jain, D., Patel, K., Brissette, P., Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", Work in Progress, Internet-Draft, draft-ietf-bess-l3vpn-yang-04, 19 October 2018, <https://tools.ietf.org/html/draft-ietf-bess-l3vpn-yang-04>.

[METRIC-METHOD] Morton, A., Geib, R., and L. Ciavattone, "Metrics and Methods for One-way IP Capacity", Work in Progress, Internet-Draft, draft-ietf-ippm-capacity-metric-method-04, 10 September 2020, <https://tools.ietf.org/html/draft-ietf-ippm-capacity-metric-method-04>.

[MVPN-YANG] Liu, Y., Guo, F., Litkowski, S., Liu, X., Kebler, R., and M. Sivakumar, "Yang Data Model for Multicast in MPLS/BGP IP VPNs", Work in Progress, Internet-Draft, draft-ietf-bess-mvpn-yang-04, 30 June 2020, <https://tools.ietf.org/html/draft-ietf-bess-mvpn-yang-04>.

[NETMOD-MODEL] Clarke, J. and B. Claise, "YANG module for yangcatalog.org", Work in Progress, Internet-Draft, draft-clacla-netmod-model-catalog-03, 3 April 2018, <https://tools.ietf.org/html/draft-clacla-netmod-model-catalog-03>.

[OPSAWG-L2NM] Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., Jalil, L., and J. Ma, "A Layer 2 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l2nm-01, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-opsawg-l2nm-01>.

[OPSAWG-L3SM-L3NM] Barguil, S., Dios, O. G. D., Boucadair, M., Munoz, L. A., and A. Aguado, "A Layer 3 VPN Network YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-l3sm-l3nm-05, 16 October 2020, <https://tools.ietf.org/html/draft-ietf-

opsawg-l3sm-l3nm-05>.

[OPSAWG-VPN-COMMON]
Barguil, S., Dios, O. G. D., Boucadair, M., and Q. Wu, "A Layer 2/3 VPN Common YANG Model", Work in Progress, Internet-Draft, draft-ietf-opsawg-vpn-common-03, 14 January 2021, <https://tools.ietf.org/html/draft-ietf-opsawg-vpn-common-03>.

[OPSAWG-YANG-VPN]
Wu, B., Wu, Q., Boucadair, M., Dios, O. G. D., Wen, B., Liu, C., and H. Xu, "A YANG Model for Network and VPN Service Performance Monitoring", Work in Progress, Internet-Draft, draft-www-opsawg-yang-vpn-service-pm-03, 21 January 2021, <https://tools.ietf.org/html/draft-www-opsawg-yang-vpn-service-pm-03>.

[PIM-YANG] Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and F. Hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", Work in Progress, Internet-Draft, draft-ietf-pim-yang-17, 19 May 2018, <https://tools.ietf.org/html/draft-ietf-pim-yang-17>.

[QOS-MODEL]
Choudhary, A., Jethanandani, M., Strahle, N., Aries, E., and I. Chen, "YANG Model for QoS", Work in Progress, Internet-Draft, draft-ietf-rtgwg-qos-model-02, 9 July 2020, <https://tools.ietf.org/html/draft-ietf-rtgwg-qos-model-02>.

[RFC4176] El Mghazli, Y., Ed., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, DOI 10.17487/RFC4176, October 2005, <https://www.rfc-editor.org/info/rfc4176>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <https://www.rfc-editor.org/info/rfc4664>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <https://www.rfc-editor.org/info/rfc4762>.

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

[RFC5486] Malas, D., Ed. and D. Meyer, Ed., "Session Peering for Multimedia Interconnect (SPEERMINT) Terminology", RFC 5486, DOI 10.17487/RFC5486, March 2009, <https://www.rfc-editor.org/info/rfc5486>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010,

<https://www.rfc-editor.org/info/rfc5880>.

[RFC6406] Malas, D., Ed. and J. Livingood, Ed., "Session PEERing for Multimedia INTerconnect (SPEERMINT) Architecture", RFC 6406, DOI 10.17487/RFC6406, November 2011, <https://www.rfc-editor.org/info/rfc6406>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <https://www.rfc-editor.org/info/rfc7224>.

[RFC7276] Mizrahi, T., Sprecher, N., Bellagamba, E., and Y. Weingarten, "An Overview of Operations, Administration, and Maintenance (OAM) Tools", RFC 7276, DOI 10.17487/RFC7276, June 2014, <https://www.rfc-editor.org/info/rfc7276>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <https://www.rfc-editor.org/info/rfc7317>.

[RFC7455] Senevirathne, T., Finn, N., Salam, S., Kumar, D., Eastlake 3rd, D., Aldrin, S., and Y. Li, "Transparent Interconnection of Lots of Links (TRILL): Fault Management", RFC 7455, DOI 10.17487/RFC7455, March 2015, <https://www.rfc-editor.org/info/rfc7455>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <https://www.rfc-editor.org/info/rfc7679>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <https://www.rfc-editor.org/info/rfc8077>.

[RFC8194] Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", RFC 8194, DOI 10.17487/RFC8194, August 2017, <https://www.rfc-editor.org/info/rfc8194>.

[RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <https://www.rfc-editor.org/info/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/info/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/info/rfc8309>.

[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/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[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/info/rfc8345>.

[RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <https://www.rfc-editor.org/info/rfc8346>.

[RFC8348] Bierman, A., Bjorklund, M., Dong, J., and D. Romascanu, "A YANG Data Model for Hardware Management", RFC 8348, DOI 10.17487/RFC8348, March 2018, <https://www.rfc-editor.org/info/rfc8348>.

[RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <https://www.rfc-editor.org/info/rfc8349>.

[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/info/rfc8466>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <https://www.rfc-editor.org/info/rfc8512>.

[RFC8513] Boucadair, M., Jacquenet, C., and S. Sivakumar, "A YANG Data Model for Dual-Stack Lite (DS-Lite)", RFC 8513, DOI 10.17487/RFC8513, January 2019, <https://www.rfc-editor.org/info/rfc8513>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <https://www.rfc-editor.org/info/rfc8519>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <https://www.rfc-editor.org/info/rfc8528>.

[RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <https://www.rfc-editor.org/info/rfc8529>.

[RFC8530] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", RFC 8530, DOI 10.17487/RFC8530, March 2019, <https://www.rfc-editor.org/info/rfc8530>.

[RFC8531] Kumar, D., Wu, Q., and Z. Wang, "Generic YANG Data Model for Connection-Oriented Operations, Administration, and Maintenance (OAM) Protocols", RFC 8531, DOI 10.17487/RFC8531, April 2019, <https://www.rfc-editor.org/info/rfc8531>.

[RFC8532] Kumar, D., Wang, Z., Wu, Q., Ed., Rahman, R., and S. Raghavan, "Generic YANG Data Model for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8532, DOI 10.17487/RFC8532, April 2019, <https://www.rfc-editor.org/info/rfc8532>.

[RFC8533] Kumar, D., Wang, M., Wu, Q., Ed., Rahman, R., and S. Raghavan, "A YANG Data Model for Retrieval Methods for the Management of Operations, Administration, and Maintenance (OAM) Protocols That Use Connectionless Communications", RFC 8533, DOI 10.17487/RFC8533, April 2019, <https://www.rfc-editor.org/info/rfc8533>.

[RFC8632] Vallin, S. and M. Bjorklund, "A YANG Data Model for Alarm Management", RFC 8632, DOI 10.17487/RFC8632, September 2019, <https://www.rfc-editor.org/info/rfc8632>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC8652] Liu, X., Guo, F., Sivakumar, M., McAllister, P., and A. Peter, "A YANG Data Model for the Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", RFC 8652, DOI 10.17487/RFC8652, November 2019, <https://www.rfc-editor.org/info/rfc8652>.

[RFC8675] Boucadair, M., Farrer, I., and R. Asati, "A YANG Data Model for Tunnel Interface Types", RFC 8675, DOI 10.17487/RFC8675, November 2019, <https://www.rfc-editor.org/info/rfc8675>.

[RFC8676] Farrer, I., Ed. and M. Boucadair, Ed., "YANG Modules for IPv4-in-IPv6 Address plus Port (A+P) Softwires", RFC 8676, DOI 10.17487/RFC8676, November 2019, <https://www.rfc-editor.org/info/rfc8676>.

[RFC8783] Boucadair, M., Ed. and T. Reddy.K, Ed., "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", RFC 8783, DOI 10.17487/RFC8783, May 2020, <https://www.rfc-editor.org/info/rfc8783>.

[RFC8791] Bierman, A., Bjrklund, M., and K. Watsen, "YANG Data Structure Extensions", RFC 8791, DOI 10.17487/RFC8791, June 2020, <https://www.rfc-editor.org/info/rfc8791>.

[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/info/rfc8795>.

[RFC8819] Hopps, C., Berger, L., and D. Bogdanovic, "YANG Module Tags", RFC 8819, DOI 10.17487/RFC8819, January 2021, <https://www.rfc-editor.org/info/rfc8819>.

[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/info/rfc8944>.

[RFC8960] Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", RFC 8960, DOI 10.17487/RFC8960, December 2020, <https://www.rfc-editor.org/info/rfc8960>.

[RTGWG-POLICY]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy", Work in Progress, Internet-Draft, draft-ietf-rtgwg-policy-model-27, 10 January 2021, <https://tools.ietf.org/html/draft-ietf-rtgwg-policy-model-27>.

[SNOOPING-YANG]
Zhao, H., Liu, X., Liu, Y., Sivakumar, M., and A. Peter, "A Yang Data Model for IGMP and MLD Snooping", Work in Progress, Internet-Draft, draft-ietf-pim-igmp-mld-snooping-yang-18, 14 August 2020, <https://tools.ietf.org/html/draft-ietf-pim-igmp-mld-snooping-yang-18>.

[SPRING-SR-YANG]
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-yang-29, 8 December 2020, <https://tools.ietf.org/html/draft-ietf-spring-sr-yang-29>.

[STAMP-YANG]
Mirsky, G., Min, X., and W. S. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", Work in Progress, Internet-Draft, draft-ietf-ippm-stamp-yang-06, 7 October 2020, <https://tools.ietf.org/html/draft-ietf-ippm-stamp-yang-06>.

[TEAS-ACTN-PM]
Lee, Y., Dhody, D., Karunanithi, S., Vilalta, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", Work in Progress, Internet-Draft, draft-ietf-teas-actn-pm-telemetry-autonomics-04, 2 November 2020, <https://tools.ietf.org/html/draft-ietf-teas-actn-pm-telemetry-autonomics-04>.

[TEAS-YANG-PATH-COMP]
Busi, I., Belotti, S., Lopez, V., Sharma, A., and Y. Shi, "Yang model for requesting Path Computation", Work in Progress, Internet-Draft, draft-ietf-teas-yang-path-computation-11, 16 November 2020, <https://tools.ietf.org/html/draft-ietf-teas-yang-path-computation-11>.

[TEAS-YANG-RSVP]
Beeram, V. P., Saad, T., Gandhi, R., Liu, X., Bryskin, I.,

and H. Shah, "A YANG Data Model for RSVP-TE Protocol", Work in Progress, Internet-Draft, draft-ietf-teas-yang-rsvp-te-08, 9 March 2020, <https://tools.ietf.org/html/draft-ietf-teas-yang-rsvp-te-08>.

[TEAS-YANG-TE]
Saad, T., Gandhi, R., Liu, X., Beeram, V. P., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels, Label Switched Paths and Interfaces", Work in Progress, Internet-Draft, draft-ietf-teas-yang-te-25, 27 July 2020, <https://tools.ietf.org/html/draft-ietf-teas-yang-te-25>.

[TRILL-YANG-OAM]
Kumar, D., Senevirathne, T., Finn, N., Salam, S., Xia, L., and W. Hao, "YANG Data Model for TRILL Operations, Administration, and Maintenance (OAM)", Work in Progress, Internet-Draft, draft-ietf-trill-yang-oam-05, 31 March 2017, <https://tools.ietf.org/html/draft-ietf-trill-yang-oam-05>.

[TWAMP-DATA-MODEL]
Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, "Two-Way Active Measurement Protocol (TWAMP) Data Model", Work in Progress, Internet-Draft, draft-ietf-ippm-twamp-yang-13, 2 July 2018, <https://tools.ietf.org/html/draft-ietf-ippm-twamp-yang-13>.

[UNI-TOPOLOGY]
Dios, O. G. D., Barguil, S., Wu, Q., and M. Boucadair, "A YANG Model for User-Network Interface (UNI) Topologies", Work in Progress, Internet-Draft, draft-ogondio-opsawg-uni-topology-01, 2 April 2020, <https://tools.ietf.org/html/draft-ogondio-opsawg-uni-topology-01>.

Appendix A. Layered YANG Module Examples Overview

This appendix lists a set of YANG data models that can be used for the delivery of connectivity services. These models can be classified as service, network, or device models.

It is not the intent of this appendix to provide an inventory of tools and mechanisms used in specific network and service management domains; such inventory can be found in documents such as [RFC7276].

The reader may refer to the YANG Catalog (<https://www.yangcatalog.org>) or the public Github YANG repository (<https://github.com/YangModels/yang>) to query existing YANG models. The YANG Catalog includes some metadata to indicate the module type ('module-classification') [NETMOD-MODEL]. Note that the mechanism defined in [RFC8819] allows to associate tags with YANG modules in order to help classifying the modules.

A.1. Service Models: Definition and Samples

As described in [RFC8309], the service is "some form of connectivity between customer sites and the Internet or between customer sites across the network operator's network and across the Internet". More concretely, an IP connectivity service can be defined as the IP transfer capability characterized by a (Source Nets, Destination Nets, Guarantees, Scope) tuple where "Source Nets" is a group of unicast IP addresses, "Destination Nets" is a group of IP unicast and/or multicast addresses, and "Guarantees" reflects the guarantees (expressed, for example, in terms of QoS, performance, and

availability) to properly forward traffic to the said "Destination" [RFC7297]. The "Scope" denotes the network perimeter (e.g., between Provider Edge (PE) routers or Customer Nodes) where the said guarantees need to be provided.

For example:

* The L3SM [RFC8299] defines the L3VPN service ordered by a customer from a network operator.

* The L2SM [RFC8466] defines the L2VPN service ordered by a customer from a network operator.

* The Virtual Network (VN) model [ACTN-VN-YANG] provides a YANG data model applicable to any mode of VN operation.

L2SM and L3SM are customer service models as per [RFC8309].

A.2. Schema Mount

Modularity and extensibility were among the leading design principles of the YANG data modeling language. As a result, the same YANG module can be combined with various sets of other modules and thus form a data model that is tailored to meet the requirements of a specific use case. [RFC8528] defines a mechanism, denoted "schema mount", that allows for mounting one data model consisting of any number of YANG modules at a specified location of another (parent) schema.

A.3. Network Models: Samples

L2NM [OPSAWG-L2NM] and L3NM [OPSAWG-L3SM-L3NM] are examples of YANG network models.

Figure 9 depicts a set of additional network models such as topology and tunnel models:

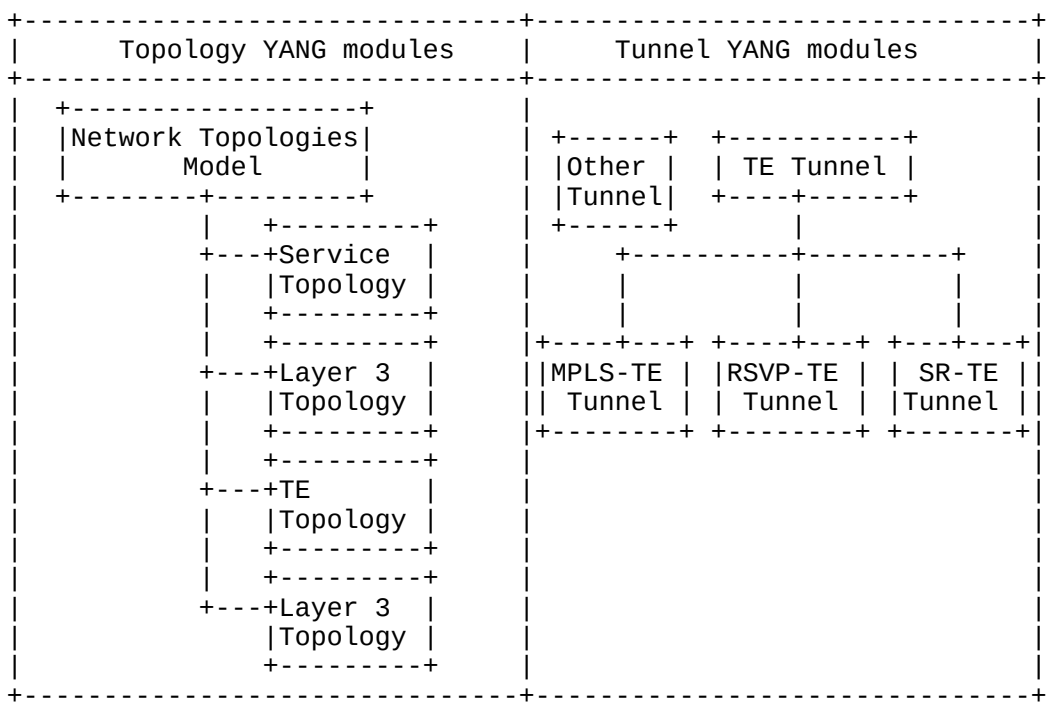


Figure 9: Sample Resource-Facing Network Models

Examples of topology YANG modules are listed below:

Network Topologies Model:
   [RFC8345] defines a base model for network topology and inventories. Network topology data includes link, node, and terminate-point resources.

TE Topology Model:
   [RFC8795] defines a YANG data model for representing and manipulating TE topologies.

   This module is extended from the network topology model defined in [RFC8345] and includes content related to TE topologies. This model contains technology-agnostic TE topology building blocks that can be augmented and used by other technology-specific TE topology models.

Layer 3 Topology Model:
   [RFC8346] defines a YANG data model for representing and manipulating Layer 3 topologies. This model is extended from the network topology model defined in [RFC8345] and includes content related to Layer 3 topology specifics.

Layer 2 Topology Model:
   [RFC8944] defines a YANG data model for representing and manipulating Layer 2 topologies. This model is extended from the network topology model defined in [RFC8345] and includes content related to Layer 2 topology specifics.

Examples of tunnel YANG modules are provided below:

Tunnel Identities:
   [RFC8675] defines a collection of YANG identities used as interface types for tunnel interfaces.

TE Tunnel Model:
   [TEAS-YANG-TE] defines a YANG module for the configuration and management of TE interfaces, tunnels, and LSPs.

Segment Routing (SR) Traffic Engineering (TE) Tunnel Model:
   [TEAS-YANG-TE] augments the TE generic and MPLS-TE model(s) and defines a YANG module for SR-TE-specific data.

MPLS-TE Model:
   [TEAS-YANG-TE] augments the TE generic and MPLS-TE model(s) and defines a YANG module for MPLS-TE configurations, state, RPC, and notifications.

RSVP-TE MPLS Model:
   [TEAS-YANG-RSVP] augments the RSVP-TE generic module with parameters to configure and manage signaling of MPLS RSVP-TE LSPs.

Other sample network models are listed hereafter:

Path Computation API Model:
   [TEAS-YANG-PATH-COMP] defines a YANG module for a stateless RPC that complements the stateful solution defined in [TEAS-YANG-TE].

OAM Models (including Fault Management (FM) and Performance Monitoring):
   [RFC8532] defines a base YANG module for the management of OAM protocols that use Connectionless Communications. [RFC8533] defines a retrieval method YANG module for connectionless OAM protocols. [RFC8531] defines a base YANG module for connection-oriented OAM protocols. These three models are intended to provide consistent reporting, configuration, and representation for connectionless OAM and connection-oriented OAM separately.

Alarm monitoring is a fundamental part of monitoring the network. Raw alarms from devices do not always tell the status of the network services or necessarily point to the root cause. [RFC8632] defines a YANG module for alarm management.

A.4. Device Models: Samples

Network Element models (listed in Figure 10) are used to describe how a service can be implemented by activating and tweaking a set of functions (enabled in one or multiple devices, or hosted in cloud infrastructures) that are involved in the service delivery. For example, the L3VPN service will involve many PEs and require manipulating the following modules:

* Routing management [RFC8349]

* BGP [IDR-BGP-MODEL]

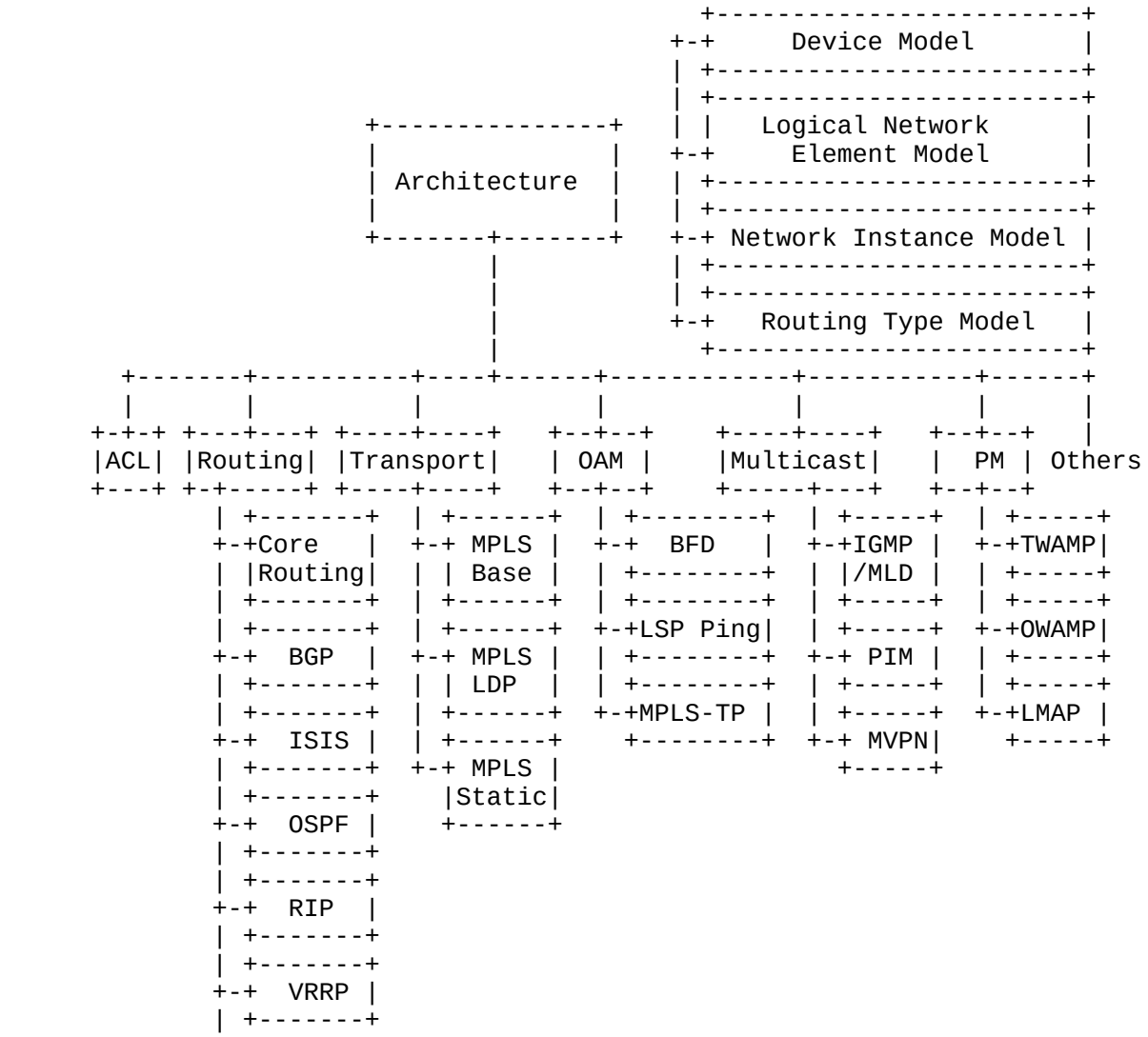
* PIM [PIM-YANG]

* NAT management [RFC8512]

* QoS management [QOS-MODEL]

* ACLs [RFC8519]

Figure 10 uses IETF-defined data models as an example.



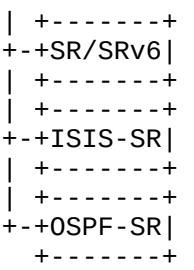


Figure 10: Network Element Models Overview

A.4.1. Model Composition

Logical Network Element Model:
[RFC8530] defines a logical network element model that can be used to manage the logical resource partitioning that may be present on a network device. Examples of common industry terms for logical resource partitioning are Logical Systems or Logical Routers.

Network Instance Model:
[RFC8529] defines a network instance module. This module can be used to manage the virtual resource partitioning that may be present on a network device. Examples of common industry terms for virtual resource partitioning are VRF instances and Virtual Switch Instances (VSIs).

A.4.2. Device Management

The following list enumerates some YANG modules that can be used for device management:

* [RFC8348] defines a YANG module for the management of hardware.

* [RFC7317] defines the "ietf-system" YANG module that provides many features such as the configuration and the monitoring of system or system control operations (e.g., shutdown, restart, and setting time) identification.

* [RFC8341] defines a network configuration access control YANG module.

A.4.3. Interface Management

The following provides some YANG modules that can be used for interface management:

* [RFC7224] defines a YANG module for interface type definitions.

* [RFC8343] defines a YANG module for the management of network interfaces.

A.4.4. Some Device Model Examples

The following provides an overview of some device models that can be used within a network. This list is not comprehensive.

L2VPN:
[L2VPN-YANG] defines a YANG module for MPLS-based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local attachment circuits. The L2VPN model covers point-to-point Virtual Private Wire Service (VPWS) and Multipoint Virtual Private LAN Service (VPLS). These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP [RFC4761].

EVPN:

[EVPN-YANG] defines a YANG module for Ethernet VPN services. The model is agnostic of the underlay. It applies to MPLS as well as to Virtual eXtensible Local Area Network (VxLAN) encapsulation. The module is also agnostic to the services, including E-LAN, E-LINE, and E-TREE services.

L3VPN:

[L3VPN-YANG] defines a YANG module that can be used to configure and manage BGP L3VPNs [RFC4364]. It contains VRF-specific parameters as well as BGP-specific parameters applicable for L3VPNs.

Core Routing:

[RFC8349] defines the core routing YANG data model, which is intended as a basis for future data model development covering more-sophisticated routing systems. It is expected that other Routing technology YANG modules (e.g., VRRP, RIP, ISIS, or OSPF models) will augment the Core Routing base YANG module.

MPLS:

[RFC8960] defines a base model for MPLS that serves as a base framework for configuring and managing an MPLS switching subsystem. It is expected that other MPLS technology YANG modules (e.g., MPLS LSP Static, LDP, or RSVP-TE models) will augment the MPLS base YANG module.

BGP:

[IDR-BGP-MODEL] defines a YANG module for configuring and managing BGP, including protocol, policy, and operational aspects based on data center, carrier, and content provider operational requirements.

Routing Policy:

[RTGWG-POLICY] defines a YANG module for configuring and managing routing policies based on operational practice. The module provides a generic policy framework that can be augmented with protocol-specific policy configuration.

SR/SRv6:

[SPRING-SR-YANG] defines a YANG module for segment routing configuration and operation.

BFD:

Bidirectional Forwarding Detection (BFD) [RFC5880] is a network protocol that is used for liveness detection of arbitrary paths between systems. [BFD-YANG] defines a YANG module that can be used to configure and manage BFD.

Multicast:

[PIM-YANG] defines a YANG module that can be used to configure and manage Protocol Independent Multicast (PIM) devices.

[RFC8652] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) devices.

[SNOOPING-YANG] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) snooping devices.

[MVPN-YANG] defines a YANG data model to configure and manage Multicast in MPLS/BGP IP VPNs (MVPNs).

PM:

[TWAMP-DATA-MODEL] defines a YANG data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP).

[STAMP-YANG] defines the data model for implementations of Session-Sender and Session-Reflector for Simple Two-way Active Measurement Protocol (STAMP) mode using YANG.

[RFC8194] defines a YANG data model for Large-Scale Measurement Platforms (LMAPs).

ACL:
An Access Control List (ACL) is one of the basic elements used to configure device-forwarding behavior. It is used in many networking technologies such as Policy-Based Routing, firewalls, etc. [RFC8519] describes a YANG data model of ACL basic building blocks.

QoS:
[QOS-MODEL] describes a YANG module of Differentiated Services for configuration and operations.

NAT:
For the sake of network automation and the need for programming the Network Address Translation (NAT) function in particular, a YANG data model for configuring and managing the NAT is essential.

[RFC8512] defines a YANG module for the NAT function covering a variety of NAT flavors such as Network Address Translation from IPv4 to IPv4 (NAT44), Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers (NAT64), customer-side translator (CLAT), Stateless IP/ICMP Translation (SIIT), Explicit Address Mappings (EAMs) for SIIT, IPv6-to-IPv6 Network Prefix Translation (NPTv6), and Destination NAT.

[RFC8513] specifies a Dual-Stack Lite (DS-Lite) YANG module.

Stateless Address Sharing:
[RFC8676] specifies a YANG module for Address plus Port (A+P) address sharing, including Lightweight 4over6, Mapping of Address and Port with Encapsulation (MAP-E), and Mapping of Address and Port using Translation (MAP-T) softwire mechanisms.

Acknowledgements

Thanks to Joe Clark, Greg Mirsky, Shunsuke Homma, Brian Carpenter, Adrian Farrel, Christian Huitema, Tommy Pauly, Ines Robles, and Olivier Augizeau for the review.

Many thanks to Robert Wilton for the detailed AD review.

Thanks to ric Vyncke, Roman Danyliw, Erik Kline, and Benjamin Kaduk for the IESG review.

Contributors

Christian Jacquenet
Orange
Rennes, 35000
France

Email: Christian.jacquenet@orange.com

Luis Miguel Contreras Murillo
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Madrid
Spain

Email: oscar.gonzalezdedios@telefonica.com

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Young Lee
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

Authors' Addresses

Qin Wu (editor)
Huawei
101 Software Avenue
Yuhua District
Nanjing
Jiangsu, 210012
China

Email: bill.wu@huawei.com

Mohamed Boucadair (editor)
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Diego R. Lopez
Telefonica I+D
Spain

Email: diego.r.lopez@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf@chinatelecom.cn

Liang Geng
China Mobile

Email: gengliang@chinamobile.com