Networking Working Group
Internet-Draft
Intended status: Informational
Expires: August 29, 2020

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

February 26, 2020

A Framework for Automating Service and Network Management with YANG
draft-ietf-opsawg-model-automation-framework-01

Abstract

Data models for service and network management provides a programmatic approach for representing (virtual) services or networks and deriving (1) configuration information that will be communicated to network and service components that are used to build and deliver the service and (2) state information that will be monitored and tracked. Indeed, data models can be used during various phases of the service and network management life cycle, such as service instantiation, service provisioning, optimization, monitoring, diagnostic, and assurance. Also, data models are instrumental in the automation of network management. They also provide closed-loop control for the sake of adaptive and deterministic service creation, delivery, and maintenance.

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

The document aims in particular to exemplify an approach that specifies the journey from technology-agnostic services to technology-specific actions.

working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 29, 2020.

Table of Contents

1. Introduction

The service management system usually comprises service activation/provision and service operation. Current service delivery procedures, from the processing of customer's requirements and order to service delivery and operation, typically assume the manipulation of data sequentially into multiple OSS/BSS applications that may be managed by different departments within the service provider's organization (e.g., billing factory, design factory, network operation center, etc.). In addition, many of these applications have been developed in-house over the years and operating in a silo mode:

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

- Secondly, many current service fulfillment system might have a limited visibility on the network state and therefore have slow response to the network changes.

Software Defined Networking (SDN) becomes crucial to address these challenges. SDN techniques [RFC7149] are meant to automate the overall service delivery procedures and typically rely upon (standard) data models that are used to not only reflect service providers'savoir-faire but also to dynamically instantiate and enforce a set of (service-inferred) policies that best accommodate what has been (contractually) defined (and possibly negotiated) with the customer. [RFC7149] provides a first tentative 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:

- 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-requirement-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 these technical items. 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 ([RFC7950]) module 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 on the top or abstracting common construct from various network technologies on the bottom. At the time of writing this document (2020), there are many 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 device, manage a set of devices involved in a service, or even provide a service is something that is not currently documented either within the IETF or other SDOs (e.g., MEF).

This document describes an architectural framework for service and network management automation (Section 3) that takes advantage of YANG modeling technologies and investigates how different layer YANG data models interact with each other (e.g., service mapping, model composing) in the context of service delivery and fulfillment (Section 4).

This framework is drawn from a network provider perspective irrespective of the origin of a data module; it can accommodate even modules that are developed outside the IETF.

The document identifies a list of use cases to exemplify the proposed approach (Section 5), but it does not claim to be exhaustive.

2. Terminology

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

- Network Operator
- Customer
- Service
- Data Model
- Service Model
- Network Element Module

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

Device Model: Refers to the Network Element YANG data module described in [RFC8199]. Device Model is also used to refer to model a function embedded in a device (e.g., NAT [RFC8512], ACL [RFC8519]).

3. Architectural Concepts & Goals

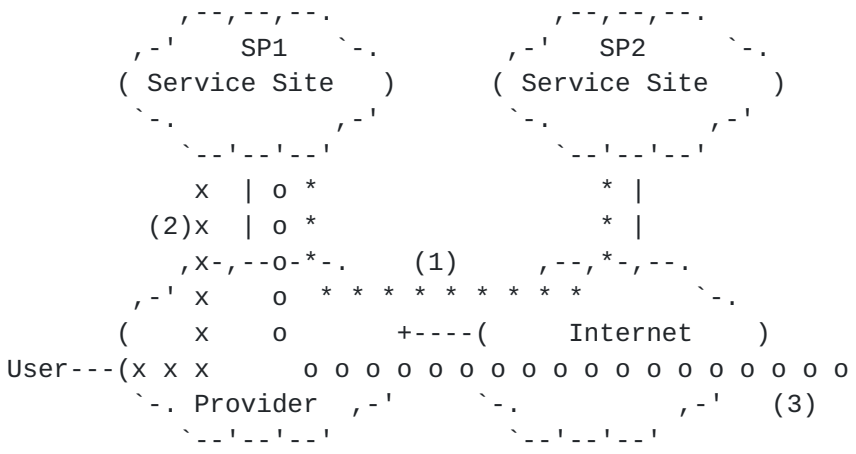
3.1. Data Models: Layering and Representation

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

The data modules 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 top down approach and are mostly customer-facing YANG modules providing a common model construct for higher level network services (e.g., L3VPN), which can be mapped to network technology-specific modules at lower layers (e.g., tunnel, routing, QoS, security). For example, the service level can be used to characterise the network service(s) to be ensured between service nodes (ingress/egress) such as the communication scope (pipe, hose, funnel, ...), the directionality, the traffic performance guarantees (one-way delay (OWD), one-way loss, ...), etc.

Figure 1 depicts the example of a VoIP service provider that relies in the connectivity services offered by a network provider. These connectivity services can be captured in a YANG Service Module 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 Connectivty Components

```
Connectivity: Scope and Guarantees
   * inter-SP connectivity (1)
      - Pipe scope from the local to the remote VoIP gateway
      - Full guarantees class
   * Customer to SP connectivity (2)
      - Hose/Funnel scope connecting the local VoIP gateway
        to the customer access points
      - Full guarantees class
   * SP to any destination connectivity (3)
      - Hose/Funnel scope from the local VoIP gateway to the
        Internet gateway
      - Delay guarantees class
Flow Identification
   * Destination IP address (SBC, SBE, DBE)
   * DSCP marking
Traffic Isolation
   * VPN
Routing & Forwarding
   * Routing rule to exclude 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 and 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, NAT).

Each level maintains a view of the supported YANG modules provided by low-levels (see for example, Appendix A).

Figure 3 illustrates the overall layering model.

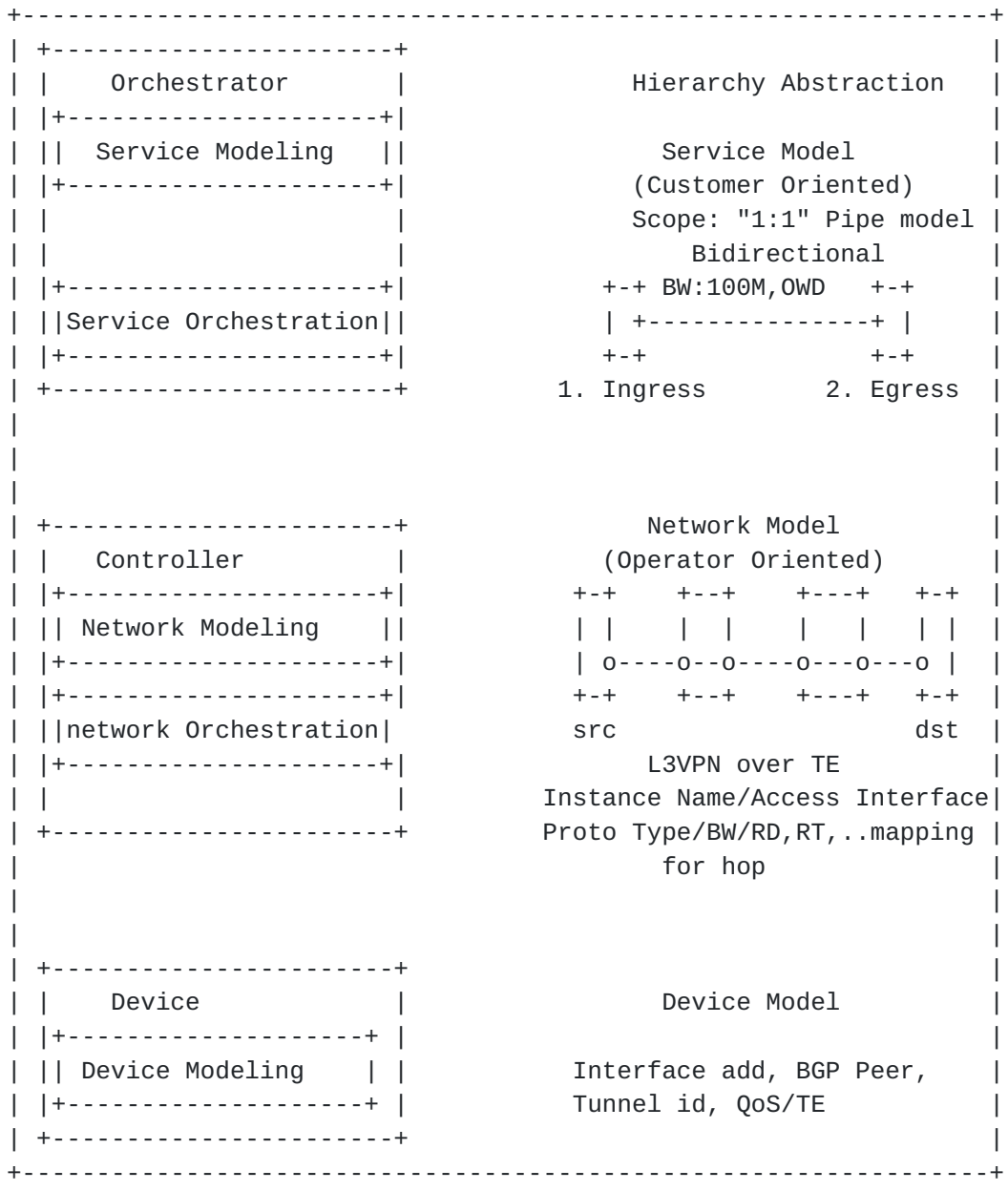


Figure 3: Layering and representation

3.2. Automation of Service Delivery Procedures

Service Models can be used by an operator to expose its services to its customers. Exposing such models allows to automate the activation and the delivery of service orders. 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 modules are used to feed a decision-making intelligence to adequately accommodate customer's needs.

Such modules may also 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 DDoS attacks [I-D.ietf-dots-signal-channel][I-D.ietf-dots-data-channel].

Network Models can be derived from Service Models and used to provision, monitor, instantiate the service, and provide lifecycle management of network resources (e.g., expose network resources to customers or operators to provide service fulfillment and assurance and allow customers or 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).

3.3. Service Fullfillment Automation

To operate a service, Device Models derived from Service Models or Network Models can be used to provision each involved network function/device with the proper configuration information, and 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 of notifications may be agreed 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 Modules Integration

To support top-down service delivery, YANG modules at different levels or at the same level need to be integrated together 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 administrative domain to support newly added module or features. A collection of Device Models integrated together can be loaded and validated during the implementation time.

High-level policies can be defined at Service or Network Models (e.g., AS 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 can trigger the generation and pushing of new telemetry data.

Performance measurement telemetry can be used to provide service assurance at Service and/or Network levels. Performance measurement telemetry model can tie with Service or Network Models to monitor network performance or Service Level Agreement.

4. Functional Bocks and Interactions

The architectural considerations described in Section 3 lead to the architecture described in this section and illustrated in Figure 4.

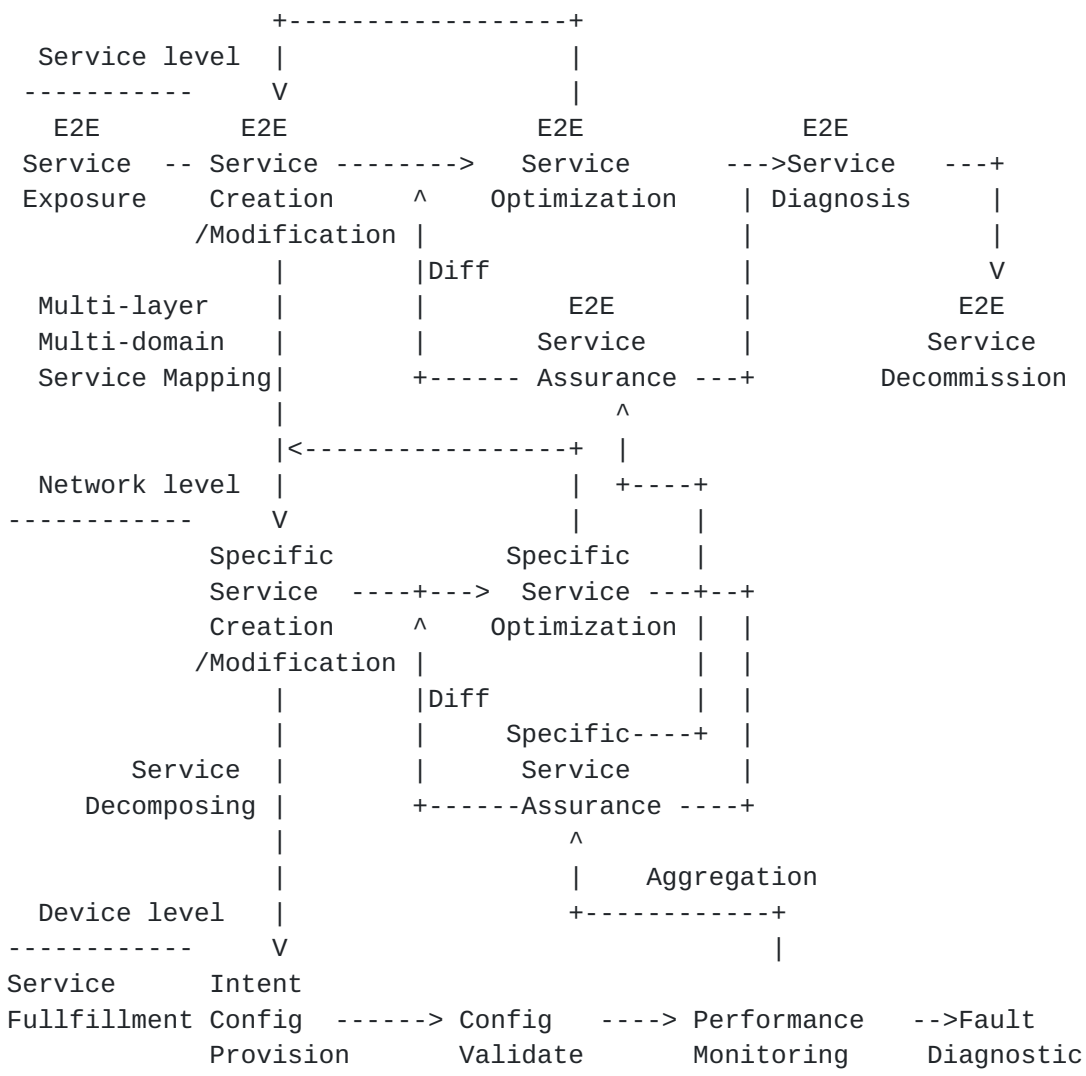


Figure 4: Service and Network Lifecycle Management

4.1. Service Lifecycle Management Procedure

Service lifecycle management includes end to end service lifecycle management at the service level and technology specific network lifecycle management at the network level. The end-to-end service lifecycle management is technology independent service management and span across multiple administrative domain or multiple layers while technology specific service lifecycle management is technology domain specific or layer specific service lifecycle management.

4.1.1. Service Exposure

A service in the context of this document (sometimes called a 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 typical example is that a customer can use a L3SM service model to request L3VPN service by providing the abstract technical characterization of the intended service between customer sites.

Service model catalogs can be created along 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 issued using the service model.

Upon receiving the service request, the service orchestrator/ management system should first verify whether the service requirements in the service request can be met (i.e., whether there is sufficient resource that can be allocated).

In successful case, the service orchestrator/management system maps such 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 to change underlying network infrastructure to adapt to new customer's needs and service requirements. This service modification can be issued in the same service model used by the service request.

4.1.3. Service Optimization

Service optimization is a technique that gets the configuration of the network updated due to network change, incident mitigation, or new service requirements. One typical example is once the tunnel or the VPN is setup, 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 modelled using YANG. With Policy-based management, self-configuration and self-optimization behavior can be specified and implemented.

4.1.4. Service Diagnosis

Operations, Administration, and Maintenance (OAM) are important networking functions for service diagnosis that allow 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 recommendation (or instructions) for the network recovery.

The service diagnosis information can be modelled 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 provide consistent configuration, reporting, and presentation for the OAM mechanisms used to manage the network.

4.1.5. Service Decommission

Service decommission allow the customer to stop the service and remove the service from active status and release the network resource that is allocated to the service. Customer can also use the service model to withdraw the registration to a service.

4.2. Service Fullfillment Management Procedure

4.2.1. Intended Configuration Provision

Intended configuration at the device level is derived from network model at the network level or service model at the service level and represents the configuration that the system attempts to apply. Take L3SM service model as an example, to deliver a L3VPN service, we need to map L3VPN service view defined in Service model into detailed

intended configuration view defined by specific configuration models for network elements, configuration information includes:

- VRF definition, including VPN Policy expression
- Physical Interface
- IP layer (IPv4, IPv6)
- QoS features such as classification, profiles, etc.
- Routing protocols: support of configuration of all protocols listed in the document, as well as routing policies associated with those protocols.
- Multicast Support
- NAT or address sharing
- Security function

This specific configuration models can be used to configure PE and CE devices within the site, e.g., a BGP policy model can be used to establish VPN membership between sites and VPN Service Topology.

4.2.2. Configuration Validation

Configuration validation is used to validate intended configuration and ensure the configuration take effect. For example, a customer creates an interface "et-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 <operational> datastore.

4.2.3. Performance Monitoring/Model-driven Telemetry

When configuration is in effect in the device, <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 does not have the visibility to the whole network or information of the flow packets are going to take through the entire network. Therefore it becomes more difficult to operate the 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 purpose and build full topological visibility to the network by aggregating and filtering these operational state from different sources.

4.2.4. Fault Diagnostic

When configuration is in effect in the device, some device may be misconfigured(e.g.,device links are not consistent on both sides of the network connection), network resources be misallocated and services may be negatively affected without knowing what is going on 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 4.1.4can be used to deal with these challenges.

These RPC commands received in the technology dependent node can be used to trigger technology specific OAM message exchange for fault verification and fault isolation,e.g., TRILL Multicast Tree Verification (MTV) RPC command [I-D.ietf-trill-yang-oam] can be used to trigger Multi-Destination Tree Verification Message defined in [RFC7455] to verify TRILL distribution tree integrity.

4.3. Multi-layer/Multi-domain Service Mapping

Multi-layer/Multi-domain Service Mapping allow you map end to end abstract view of the service segmented at different layer or different administrative domain into domain specific view. One example is to map service parameters in L3VPN service model into configuration parameters such as RD, RT, and VRF in L3VPN network model. Another example is to map service parameters in L3VPN service model into TE tunnel parameter (e.g.,Tunnel ID) in TE model and VN parameters (e.g., AP list, VN member) in TEAS VN model [I-D.ietf-teas-actn-vn-yang].

4.4. Service Decomposing

Service Decomposing allows to decompose service model at the service level or network model at the network level into a set of device/ function models at the device level. These device models may be tied to specific device type or classified into a collection of related YANG modules based on service type and feature offered and load at the implementation time before configuration is loaded and validated.

5. YANG Data Model Integration Examples

5.1. L3VPN Service Delivery

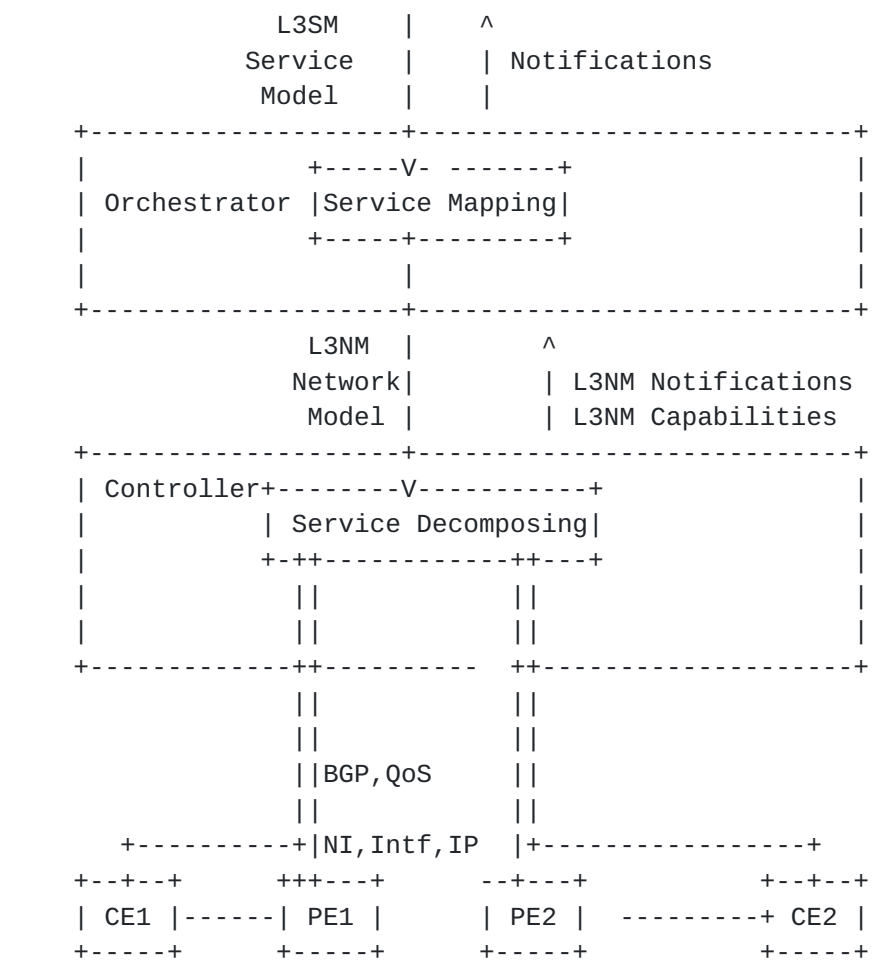


Figure 5: L3VPN Service Delivery Example

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

1. The Customer requests to create two sites (as per service creation operation in Section 4.2.1) relying upon a L3SM Service model with each having one network access connectivity:

   Site A: Network-Access A, Bandwidth=20M, for class "foo", guaranteed-bw-percent = 10, One-Way-Delay=70 msec

   Site B: Network-Access B, Bandwidth=30M, for class "foo1", guaranteed-bw-percent = 15, One-Way-Delay=60 msec

2. The Orchestrator extracts the service parameters from the L3SM model. Then, it uses them as input to translate ("service mapping operation" in Section 4.4) them into an orchestrated

configuration of network elements (e.g., RD, RT, VRF) that are part of the L3NM network model.

3. The Controller takes orchestrated configuration parameters in the L3NM network model and translates them into orchestrated ("service decomposing operation" in) configuration of network elements that are part of, e.g, BGP, QoS, Network Instance model, IP management, and interface models.

[I-D.ogondio-opsawg-uni-topology] is used for representing, managing and controlling the User Network Interface (UNI) topology.

L3NM inherits some of data elements from the L3SM. Likewise, the L3NM 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).

5.2. VN Lifecycle Management

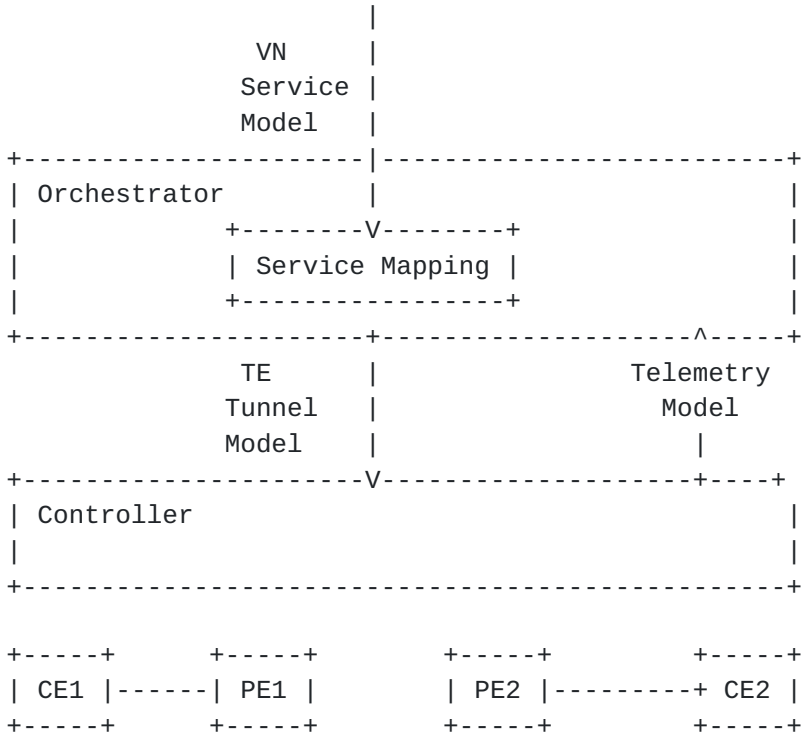


Figure 6

In reference to Figure 6, the following steps are performed to deliver the VN service within the network management automation architecture defined in this document:

1. Customer requests (service exposure operation in Section 4.1.1) to create 'VN' based on Access point, association between VN and Access point, VN member defined in the VN YANG module.

2. The orchestrator creates the single abstract node topology based on the information captured in an VN YANG module.

3. The Customer exchanges connectivity-matrix on abstract node and explicit path using TE topology model with the orchestrator. This information can be used to instantiate VN and setup tunnels between source and destination endpoints (service creation operation in Section 4.1.2).

4. The telemetry model which augments the TEAS VN model and corresponding TE Tunnel model can be used to subscribe to performance measurement data and notify all the parameter changes and network performance change related to VN topology or Tunnel [I-D.ietf-teas-actn-pm-telemetry-autonomics] and provide service assurance (service optimization operation in Section 4.1.3).

5.3. Event-based Telemetry in the Device Self management

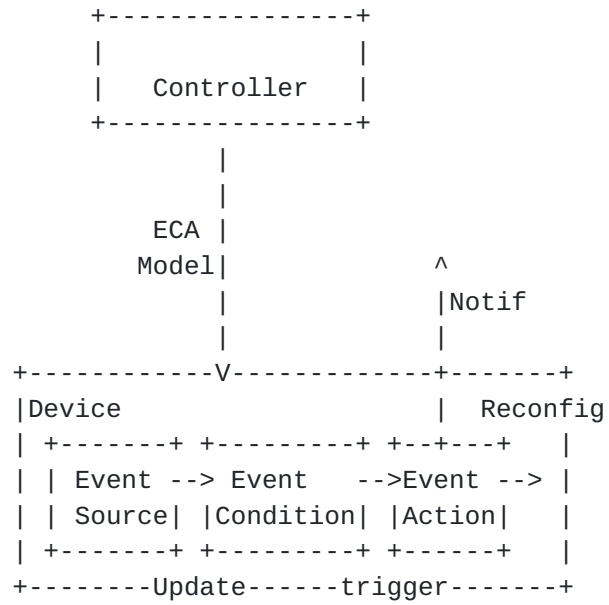


Figure 7: Event-based Telemetry

In reference to Figure 7, the following steps are performed to monitor state changes of managed objects or resource in the device and provide device self management within the network management automation architecture defined in this document:

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 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-driven policy or network control logic in the controller.

2. The controller pushes ECA policy to the network device and delegate network control logic to the network device.

3. The network device generates ECA script from ECA model and execute ECA script or network control logic based on Event. Event based notification or telemetry can be triggered if a certain condition is satisfied (model driven telemetry operation in Section 4.2.3).

6. Security Considerations

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

(Potential) security considerations specific to this document are listed below:

- Create forwarding loops by mis-configuring the underlying network.

- Leak sensitive information: special care should be considered when translating between the various layers introduced in the document.

- Some Service Models may include a traffic isolation clause, appropriate technology-specific actions must be enforced to avoid that traffic is accessible to non-authorized parties.

7. IANA Considerations

There are no IANA requests or assignments included in this document.

8. Acknowledgements

Thanks to Joe Clark, Greg Mirsky, and Shunsuke Homma for the review.

9. Contributors

Christian Jacquenet
Orange
Rennes, 35000
France
Email: Christian.jacquenet@orange.com

Luis Miguel Contreras Murillo
Telifonica

Email: luismiguel.contrerasmurillo@telefonica.com

Oscar Gonzalez de Dios
Telefonica
Madrid
ES

Email: oscar.gonzalezdedios@telefonica.com

Chongfeng Xie
China Telecom
Beijing
China

Email: xiechf.bri@chinatelecom.cn

Weiqiang Cheng
China Mobile

Email: chengweiqiang@chinamobile.com

Young Lee
Sung Kyun Kwan University

Email: younglee.tx@gmail.com

10. References

10.1. Normative References

[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>.

10.2. Informative References

[I-D.arkko-arch-virtualization]
Arkko, J., Tantsura, J., Halpern, J., and B. Varga, "Considerations on Network Virtualization and Slicing", draft-arkko-arch-virtualization-01 (work in progress), March 2018.

[I-D.asechoud-netmod-diffserv-model]
Choudhary, A., Shah, S., Jethanandani, M., Liu, B., and N. Strahle, "YANG Model for Diffserv", draft-asechoud-netmod-diffserv-model-03 (work in progress), June 2015.

[I-D.clacla-netmod-model-catalog]
Clarke, J. and B. Claise, "YANG module for yangcatalog.org", draft-clacla-netmod-model-catalog-03 (work in progress), April 2018.

[I-D.homma-slice-provision-models]
Homma, S., Nishihara, H., Miyasaka, T., Galis, A., OV, V., Lopez, D., Contreras, L., Ordonez-Lucena, J., Martinez-Julia, P., Qiang, L., Rokui, R., Ciavaglia, L., and X. Foy, "Network Slice Provision Models", draft-homma-slice-provision-models-02 (work in progress), November 2019.

[I-D.ietf-bess-evpn-yang]
Brissette, P., Shah, H., Hussain, I., Tiruveedhula, K., and J. Rabadan, "Yang Data Model for EVPN", draft-ietf-bess-evpn-yang-07 (work in progress), March 2019.

[I-D.ietf-bess-l2vpn-yang]
Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-10 (work in progress), July 2019.

[I-D.ietf-bess-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", draft-ietf-bess-l3vpn-yang-04 (work in progress), October 2018.

[I-D.ietf-bfd-yang]
Rahman, R., Zheng, L., Jethanandani, M., Pallagatti, S., and G. Mirsky, "YANG Data Model for Bidirectional Forwarding Detection (BFD)", draft-ietf-bfd-yang-17 (work in progress), August 2018.

[I-D.ietf-ccamp-alarm-module]
Vallin, S. and M. Bjorklund, "YANG Alarm Module", draft-ietf-ccamp-alarm-module-09 (work in progress), April 2019.

[I-D.ietf-ccamp-flexigrid-media-channel-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., Dios, O., King, D., Lee, Y., and G. Galimberti, "YANG data model for Flexi-Grid media-channels", draft-ietf-ccamp-flexigrid-media-channel-yang-02 (work in progress), March 2019.

[I-D.ietf-ccamp-flexigrid-yang]
Madrid, U., Perdices, D., Lopezalvarez, V., King, D., Lee, Y., and H. Zheng, "YANG data model for Flexi-Grid Optical Networks", draft-ietf-ccamp-flexigrid-yang-05 (work in progress), January 2020.

[I-D.ietf-ccamp-l1csm-yang]
Lee, Y., Lee, K., Zheng, H., Dhody, D., Dios, O., and D. Ceccarelli, "A YANG Data Model for L1 Connectivity Service Model (L1CSM)", draft-ietf-ccamp-l1csm-yang-10 (work in progress), September 2019.

[I-D.ietf-ccamp-mw-yang]
Ahlberg, J., Ye, M., Li, X., Spreafico, D., and M. Vaupotic, "A YANG Data Model for Microwave Radio Link", draft-ietf-ccamp-mw-yang-13 (work in progress), November 2018.

[I-D.ietf-ccamp-otn-topo-yang]
Zheng, H., Busi, I., Liu, X., Belotti, S., and O. Dios, "A YANG Data Model for Optical Transport Network Topology", draft-ietf-ccamp-otn-topo-yang-09 (work in progress), November 2019.

[I-D.ietf-ccamp-otn-tunnel-model]
Zheng, H., Busi, I., Belotti, S., Lopezalvarez, V., and Y. Xu, "OTN Tunnel YANG Model", draft-ietf-ccamp-otn-tunnel-model-09 (work in progress), November 2019.

[I-D.ietf-ccamp-wson-tunnel-model]
Lee, Y., Zheng, H., Guo, A., Lopezalvarez, V., King, D., Yoon, B., and R. Vilata, "A Yang Data Model for WSON Tunnel", draft-ietf-ccamp-wson-tunnel-model-04 (work in progress), September 2019.

[I-D.ietf-dots-data-channel]
Boucadair, M. and T. Reddy.K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", draft-ietf-dots-data-channel-31 (work in progress), July 2019.

[I-D.ietf-dots-signal-channel]
Reddy.K, T., Boucadair, M., Patil, P., Mortensen, A., and N. Teague, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", draft-ietf-dots-signal-channel-41 (work in progress), January 2020.

[I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-07 (work in progress), October 2019.

[I-D.ietf-ippm-stamp-yang]
Mirsky, G., Xiao, M., and W. Luo, "Simple Two-way Active Measurement Protocol (STAMP) Data Model", draft-ietf-ippm-stamp-yang-05 (work in progress), October 2019.

[I-D.ietf-ippm-twamp-yang]
Civil, R., Morton, A., Rahman, R., Jethanandani, M., and K. Pentikousis, "Two-Way Active Measurement Protocol (TWAMP) Data Model", draft-ietf-ippm-twamp-yang-13 (work in progress), July 2018.

[I-D.ietf-mpls-base-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., and V. Beeram, "A YANG Data Model for MPLS Base", draft-ietf-mpls-base-yang-12 (work in progress), February 2020.

[I-D.ietf-pim-igmp-mld-snooping-yang]
Zhao, H., Liu, X., Liu, Y., Sivakumar, M., and A. Peter, "A Yang Data Model for IGMP and MLD Snooping", draft-ietf-pim-igmp-mld-snooping-yang-09 (work in progress), January 2020.

[I-D.ietf-pim-igmp-mld-yang]
Liu, X., Guo, F., Sivakumar, M., McAllister, P., and A. Peter, "A YANG Data Model for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD)", draft-ietf-pim-igmp-mld-yang-15 (work in progress), June 2019.

[I-D.ietf-pim-yang]
Liu, X., McAllister, P., Peter, A., Sivakumar, M., Liu, Y., and f. hu, "A YANG Data Model for Protocol Independent Multicast (PIM)", draft-ietf-pim-yang-17 (work in progress), May 2018.

[I-D.ietf-rtgwg-device-model]
Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", draft-ietf-rtgwg-device-model-02 (work in progress), March 2017.

[I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy Management", draft-ietf-rtgwg-policy-model-08 (work in progress), January 2020.

[I-D.ietf-softwire-iftunnel]
Boucadair, M., Farrer, I., and R. Asati, "Tunnel Interface Types YANG Module", draft-ietf-softwire-iftunnel-07 (work in progress), June 2019.

[I-D.ietf-softwire-yang]
Farrer, I. and M. Boucadair, "YANG Modules for IPv4-in-IPv6 Address plus Port (A+P) Softwires", draft-ietf-softwire-yang-16 (work in progress), January 2019.

[I-D.ietf-spring-sr-yang]
Litkowski, S., Qu, Y., Lindem, A., Sarkar, P., and J. Tantsura, "YANG Data Model for Segment Routing", draft-ietf-spring-sr-yang-15 (work in progress), January 2020.

[I-D.ietf-supa-generic-policy-data-model]
Halpern, J. and J. Strassner, "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)", draft-ietf-supa-generic-policy-data-model-04 (work in progress), June 2017.

[I-D.ietf-teas-actn-pm-telemetry-autonomics]
Lee, Y., Dhody, D., Karunanithi, S., Vilata, R., King, D., and D. Ceccarelli, "YANG models for VN/TE Performance Monitoring Telemetry and Scaling Intent Autonomics", draft-ietf-teas-actn-pm-telemetry-autonomics-01 (work in progress), October 2019.

[I-D.ietf-teas-actn-vn-yang]
Lee, Y., Dhody, D., Ceccarelli, D., Bryskin, I., and B. Yoon, "A Yang Data Model for VN Operation", draft-ietf-teas-actn-vn-yang-07 (work in progress), October 2019.

[I-D.ietf-teas-sf-aware-topo-model]
Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., and J. Tantsura, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-04 (work in progress), November 2019.

[I-D.ietf-teas-te-service-mapping-yang]
Lee, Y., Dhody, D., Fioccola, G., WU, Q., Ceccarelli, D., and J. Tantsura, "Traffic Engineering (TE) and Service Mapping Yang Model", draft-ietf-teas-te-service-mapping-yang-02 (work in progress), September 2019.

[I-D.ietf-teas-yang-l3-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Layer 3 TE Topologies", draft-ietf-teas-yang-l3-te-topo-05 (work in progress), July 2019.

[I-D.ietf-teas-yang-path-computation]
Busi, I., Belotti, S., Lopezalvarez, V., Sharma, A., and Y. Shi, "Yang model for requesting Path Computation", draft-ietf-teas-yang-path-computation-08 (work in progress), December 2019.

[I-D.ietf-teas-yang-rsvp-te]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Bryskin, I., and H. Shah, "A YANG Data Model for RSVP-TE Protocol", draft-ietf-teas-yang-rsvp-te-07 (work in progress), July 2019.

[I-D.ietf-teas-yang-sr-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and S. Litkowski, "YANG Data Model for SR and SR TE Topologies", draft-ietf-teas-yang-sr-te-topo-06 (work in progress), November 2019.

[I-D.ietf-teas-yang-te]
Saad, T., Gandhi, R., Liu, X., Beeram, V., and I. Bryskin, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces", draft-ietf-teas-yang-te-22 (work in progress), November 2019.

[I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for Traffic Engineering (TE) Topologies", draft-ietf-teas-yang-te-topo-22 (work in progress), June 2019.

[I-D.ietf-trill-yang-oam]
Kumar, D., Senevirathne, T., Finn, N., Salam, S., Xia, L., and H. Weiguo, "YANG Data Model for TRILL Operations, Administration, and Maintenance (OAM)", draft-ietf-trill-yang-oam-05 (work in progress), March 2017.

[I-D.ogondio-opsawg-uni-topology]
Dios, O., Barguil, S., WU, Q., and M. Boucadair, "A YANG Model for User-Network Interface (UNI) Topologies", draft-ogondio-opsawg-uni-topology-00 (work in progress), November 2019.

[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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC8328] Liu, W., Xie, C., Strassner, J., Karagiannis, G., Klyus, M., Bi, J., Cheng, Y., and D. Zhang, "Policy-Based Management Framework for the Simplified Use of Policy Abstractions (SUPA)", RFC 8328, DOI 10.17487/RFC8328, March 2018, <https://www.rfc-editor.org/info/rfc8328>.

[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>.

[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>.

[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>.

Appendix A. Layered YANG Modules Example Overview

It is not the intent of this document 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].

A.1. Service Models: Definition and Samples

As described in [RFC8309], the service is "some form of connectivity between customer sites and the Internet and/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 in terms of Quality Of Service (QoS), performance, and availability, for example) to properly forward traffic to the said "Destination" [RFC7297].

For example:

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

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

- VN model [I-D.ietf-teas-actn-vn-yang]provides a YANG data model generally applicable to any mode of Virtual Network (VN) operation.

A.2. Network Models: Definitions and Samples

Figure 8 depicts a set of Network models such as topology models or tunnel models:

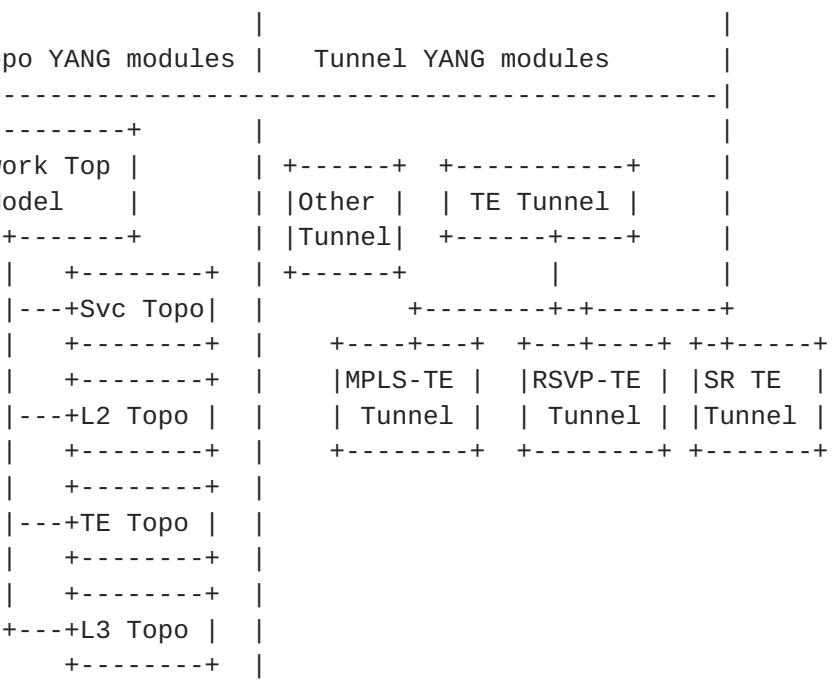


Figure 8: Sample Resource Facing Network Models

Topology YANG module Examples:

- Network Topology Models: [RFC8345] defines a base model for network topology and inventories. Network topology data include link resource, node resource, and terminate-point resources.

- TE Topology Models: [I.D-ietf-teas-yang-te-topo] defines a data model for representing and manipulating TE topologies.

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

- L3 Topology Models

  [RFC8346] defines a data model for representing and manipulating L3 Topologies. This model is extended from the network topology model defined in [RFC8345] with L3 topologies specifics.

- L2 Topology Models

  [I.D-ietf-i2rs-yang-l2-topology] defines a data model for representing and manipulating L2 Topologies. This model is extended from the network topology model defined in [RFC8345] with L2 topologies specifics.

Tunnel YANG module Examples:

- Tunnel identities [I-D.ietf-softwire-iftunnel] to ease manipulating extensions to specific tunnels.

- TE Tunnel Model

  [I.D-ietf-teas-yang-te] defines a YANG module for the configuration and management of TE interfaces, tunnels and LSPs.

- SR TE Tunnel Model

  [I.D-ietf-teas-yang-te] augments the TE generic and MPLS-TE model(s) and defines a YANG module for Segment Routing (SR) TE specific data.

- MPLS TE Model

  [I.D-ietf-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

  [I.D-ietf-teas-yang-rsvp-te] augments the RSVP-TE generic module with parameters to configure and manage signaling of MPLS RSVP-TE LSPs.

Other Network Models:

- Path Computation API Model

  [I.D-ietf-teas-path-computation] YANG module for a stateless RPC which complements the stateful solution defined in [I.D-ietf-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 connection-less 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. [I.D-ietf-ccamp-alarm-module] defines a YANG module for alarm management.

- Generic Policy Model

The Simplified Use of Policy Abstractions (SUPA) policy-based management framework [RFC8328] defines base YANG modules [I-D.ietf-supa-generic-policy-data-model]to encode policy. These models point to device-, technology-, and service-specific YANG modules developed elsewhere. Policy rules within an operator's environment can be used to express high-level, possibly network-wide, policies to a network management function (within a controller, an orchestrator, or a network element). The network management function can then control the configuration and/or monitoring of network elements and services. This document describes the SUPA basic framework, its elements, and interfaces.

A.3. Device Models: Definitions and Samples

Network Element models (Figure 9) 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. The following figure uses IETF defined models as an example.

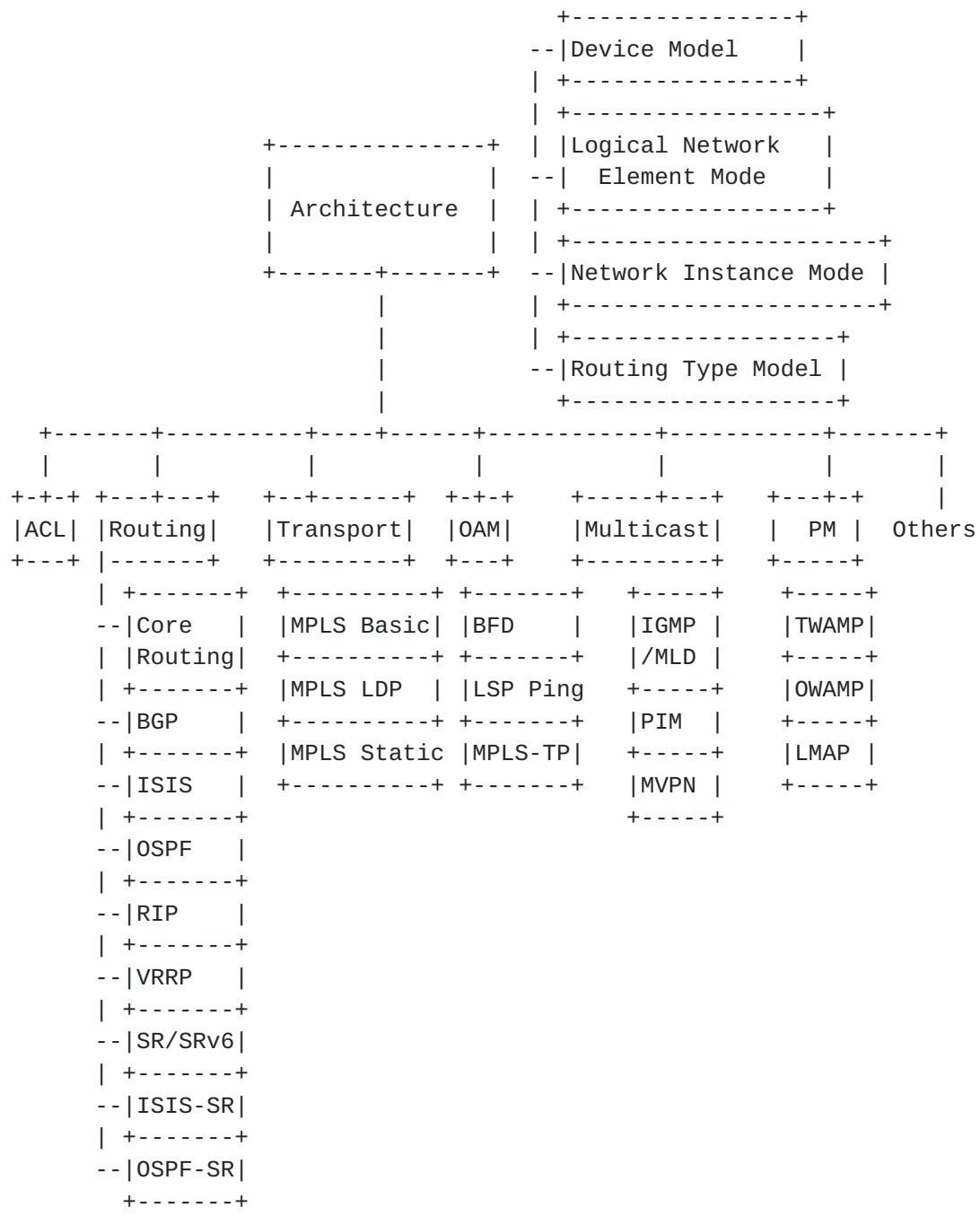


Figure 9: Network Element Modules Overview

A.3.1. Model Composition

- Device Model

  [I.D-ietf-rtgwg-device-model] presents an approach for organizing YANG modules in a comprehensive logical structure that may be used to configure and operate network devices. The structure is itself

represented as an example YANG module, with all of the related component models logically organized in a way that is operationally intuitive, but this model is not expected to be implemented.

- Logical Network Element Model

  [RFC8530] defines a logical network element module which 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 Virtual Routing and Forwarding (VRF) instances and Virtual Switch Instances (VSIs).

A.3.1.1. 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.

That capability does not cover design time.

A.3.2. Device Models: Definitions and Samples

BGP: [I-D.ietf-idr-bgp-yang-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.

MPLS: [I-D.ietf-mpls-base-yang] defines a base model for MPLS which 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.

QoS: [I-D.asechoud-netmod-diffserv-model] describes a YANG module of Differentiated Services for configuration and operations.

ACL: 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 data model of Access Control List (ACL) basic building blocks.

NAT: For the sake of network automation and the need for programming Network Address Translation (NAT) function in particular, a 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 (EAM) for SIIT, IPv6-to-IPv6 Network Prefix Translation (NPTv6), and Destination NAT. [RFC8513] specifies a YANG module for the DS-Lite AFTR.

Stateless Address Sharing: [I-D.ietf-softwire-yang] specifies a YANG module for 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.

Multicast: [I-D.ietf-pim-yang] defines a YANG module that can be used to configure and manage Protocol Independent Multicast (PIM) devices. [I-D.ietf-pim-igmp-mld-yang] defines a YANG module that can be used to configure and manage Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) devices. [I-D.ietf-pim-igmp-mld-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.

EVPN: [I-D.ietf-bess-evpn-yang] defines a YANG module for Ethernet VPN services. The model is agnostic of the underlay. It apply to MPLS as well as to VxLAN encapsulation. The model is also agnostic of the services including E-LAN, E-LINE and E-TREE services. This document mainly focuses on EVPN and Ethernet-Segment instance framework.

L3VPN: [I-D.ietf-bess-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.

L2VPN: [I-D.ietf-bess-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 VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP [RFC4761].

Routing Policy: [I-D.ietf-rtgwg-policy-model] defines a YANG module for configuring and managing routing policies in a vendor-neutral way and based on actual operational practice. The model provides a generic policy framework which can be augmented with protocol-specific policy configuration.

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

SR/SRv6: [I-D.ietf-spring-sr-yang] a YANG module for segment routing configuration and operation. [I-D.raza-spring-srv6-yang] defines a YANG module for Segment Routing IPv6 (SRv6) base. The model serves as a base framework for configuring and managing an SRv6 subsystem and expected to be augmented by other SRv6 technology models accordingly.

Core Routing: [RFC8349] defines the core routing 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, OSPF models) will augment the Core Routing base YANG module.

PM:

[I.D-ietf-ippm-twamp-yang] defines a data model for client and server implementations of the Two-Way Active Measurement Protocol (TWAMP).

[I.D-ietf-ippm-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 data model for Large-Scale Measurement Platforms (LMAPs).

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.bri@chinatelecom.cn

Liang Geng
China Mobile

Email: gengliang@chinamobile.com