

Onsen Working Group
Internet-Draft
Intended status: Informational
Expires: 8 August 2026

C. Xie
Q. Sun
China Telecom
B. Claise
Everything-OPS
L. Dunbar
Futurewei
LM. Contreras
Telefonica
K. Lambrechts
Netedge
B. Wu
Huawei
4 February 2026

ONSEN Problem Statement
draft-xie-onsen-problem-statement-00

Abstract

Exposure of network and service abstractions to external systems via YANG-based APIs has become an important means for operators to enhance their competitiveness. Despite the availability of numerous YANG data models, operators continue to face significant challenges in operationalizing these APIs in a consistent, scalable, and interoperable manner. Based on a typical use case, this document describes the problem space associated with operationalizing YANG-based service APIs, drawing on operator experience, IAB workshop findings to motivate requirements for improving API predictability and operational effectiveness through the upcoming ONSSEN WG. In addition, it intends to gather operational needs and motivations for network and service abstractions to inform YANG data model refresh work.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute 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 8 August 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
2. Terminology	4
3. One Typical Use Case Highlighting Challenges	5
3.1. Overview	5
3.2. Overall Composition of the System	5
3.3. Workflow	7
3.4. APIs to External Systems	9
3.5. Issues Identified	11
4. Operational Challenges with Network and Service Abstractions	13
4.1. Fragmented Operational Lifecycles	13
4.2. Misalignment Between Abstraction Layers	14
4.3. Inconsistent Semantics and Operational Assumptions	14
4.4. Limited Feedback and Observability for Abstraction Enforcement	15
4.5. Impact on Operational Efficiency and Interoperability	15
5. Operational Evidence from the IAB NEMOPS Workshop	15
6. Operational Needs Highlighted by the Use Cases	16
7. Operational Considerations	17
8. Security Considerations	18
9. IANA Considerations	18
10. References	18
10.1. Normative References	18
10.2. Informative References	19

Authors' Addresses	20
------------------------------	----

1. Introduction

Despite the availability of numerous YANG data models, operators continue to face significant challenges in operationalizing YANG-based service APIs in a consistent, scalable, and interoperable manner. As highlighted by the IAB Next Era of Network Management Operations [NEMOPS] workshop, operational workflows that rely on these APIs remain fragmented and difficult to automate end-to-end. In practice, APIs generated from similar YANG models often differ in service semantics, complicating integration across systems, vendors, and deployment environments. In this document, service semantics refers to the operational meaning of service abstractions as exposed via YANG-based APIs, such as lifecycle behavior, validity, duration, and feedback, rather than YANG syntax itself.

Operationalizing Network and Service abstractionNs (ONSEN) Working Group is chartered to address this problem space by focusing on the operational aspects of YANG-based service APIs, rather than defining new protocols or API technologies. The goal of ONSEN is to improve automation, operational efficiency, and interoperability by identifying common problems, clarifying requirements, and providing guidance on how YANG-based service APIs should be structured, exposed, and consumed by external systems in a predictable and interoperable manner.

This document aims to gather operational needs and motivations for network and service abstractions to inform YANG data model refresh work. It introduces a use case of data transmission service for data-intensive workload, which supports ultra-high speed, deterministic time periods, and consists of multiple segments. For the service provisioning over multi-domain heterogeneous networks, the network orchestrator exposed specialized APIs to BSS or external systems, and based on practice, designing a new service model has been identified. Model refresh work has been manifested by [I-D.bg-onions-update-network-service-models], which provides the findings from the implementations, deriving the functionalities required to update the Service and Network YANG data models. In addition, it consolidates operator-observed challenges and problems related to YANG-based service APIs, explains why existing approaches and tools are insufficient when considered in isolation, and frames the requirements that ONSEN is chartered to examine to improve the operationalization and consumption of YANG-based service APIs. This document does not propose specific solutions, protocols, or data models.

1.1. Requirements Language

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

2. Terminology

The following terms are used in this document:

- * AC: Attachment Circuit.
- * API: Application Programming Interface.
- * Abstraction: refers to the definition of simplified, high-level constructs that represent network and service capabilities, while hiding the details of their underlying realization. Such abstractions enable interaction between management and automation systems without requiring direct exposure of device-specific configurations or protocol behaviors.
- * BSS: Business Support Systems.
- * BR: Border Router.
- * CLI: Command-Line Interface.
- * CRM: Customer Relation Managment.
- * CPE: Customer Premise Edge
- * DTS-I: Data Transmission Service for data-Intensive workload.
- * NEMOPS: Next Era of Network Management Operations.
- * ONSEN: Operationalizing Network and Service abstractionNs.
- * OSS: Operation Support Systems.
- * PPPoEo6: PPPoE over IPv6.
- * SGW: Service Gateway.
- * SNMP: Simple Network Management Protocol.

3. One Typical Use Case Highlighting Challenges

The IETF has produced several YANG data models that are instrumental for automating the provisioning and delivery of connectivity services, they are described in [RFC8969]. They can be used in various use cases, such as inter-data-center connectivity and on-orbit networking with dynamic interconnection. In the latter environment, network services must be instantiated, modified, and torn down on short timescales. Constructs such as bearers or attachment circuits may need to be dynamically established to interconnect infrastructure at orbital speeds. YANG-based service APIs are a natural interface for exposing such services to planning and control systems. To further elaborate on the challenges operators face in operationalizing YANG-based APIs, a new use case with the name of "Data Transmission Service for Data-Intensive Workload(in short, DTS-I)" is illustrated by this section.

3.1. Overview

With the advent of the AI era, customers such as scientific research, education, and biotechnology have an increasing need to transfer burst-like massive data. In this context, massive datasets are routinely transferred to centralized computing and storage infrastructure for analysis and processing. These transfers may involve large volumes of data, require predictable completion times, and demand bandwidth that varies significantly over time. To meet this demand, operators need support service automation configuration, enabling elastic high-bandwidth, mission-oriented, and rapid (within seconds) network creation across heterogeneous networks.

3.2. Overall Composition of the System

This use case follows the rationale described in [RFC8969]. Figure 1 illustrates a network example connecting multiple data centers (DCs) and enterprise CPEs, the network may span multiple domains or even multiple operators.

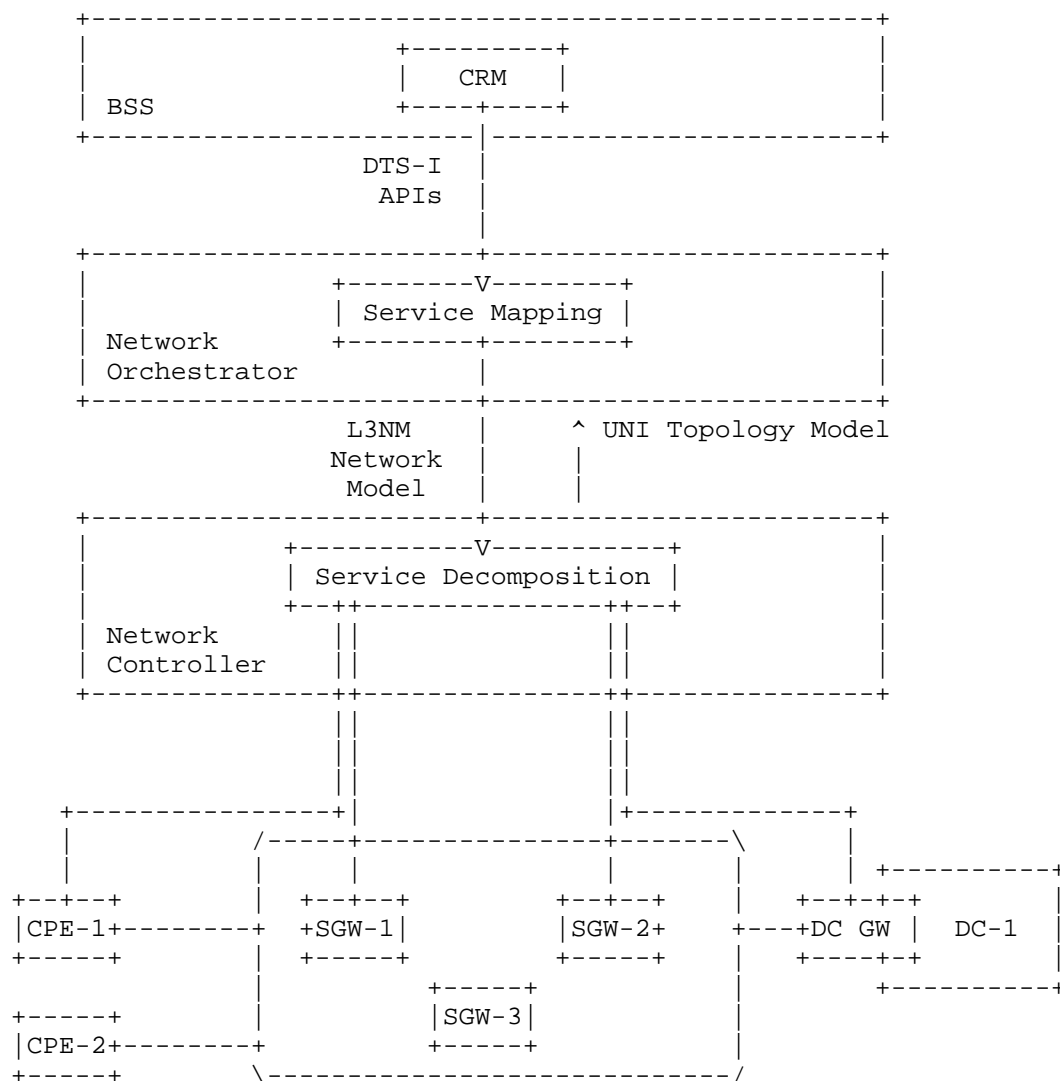


Figure 1: DTS-I Service Delivery Example

Each DC potentially hosts different computation resources. For instance, DC-1 is directly linked to the network, suggesting it host sufficient computing and storage capabilities. Various DC gateways (DC GWs) are deployed to manage traffic flow towards DCs. Two CPE devices (CPE-1 of user A and CPE-2 of User B) are connected to edge of the network respectively, indicating the start points for customer traffic entering the provider's network. There are several SGW

devices (SGW-1, SGW-2, and SGW-3, etc.) which serve as entry and exit points for massive data transmission between the customer and the DCs. The Border routers (BRs) facilitate the inter-connection of different parts of the network. They connect the access/aggregation layer to the core network.

When deploying DTS-I services by creating overlay connection for scheduled data transmission across this network, generic API calls are needed. A typical usage is to use Restful APIs of Network Orchestrator to provision the DTS-I connection from CPE-1 to DC-1. In this case, the DTS-I connection consists of three segments: the access segment from CPE-1 to SGW-1, the VPN segment from SGW-1 to SGW-2, and the segment between SGW-2 to the DC-1 egress point. Therefore, this is a composite connection.

In this case, it is assumed that the Network Orchestrator has access to the DC and network connectivity topology (e.g. TE Topology [RFC8975]), as well as resource(i.e., bandwidth) information for the network. Some standard network inventory interfaces are available. For example, the Service Attachment Points (SAPs) [RFC9408] or Acs [RFC9834] can obtain the AC/Bearer information of the PE, which suggests that the private line service provisioning resource resources on the network side.

3.3. Workflow

The workflow below gives an example of efficient use of network connections for massive data transmission in this case.

Step 1: Service ordering

Via the Service Portal on the BSS, the customer input the relevant information into the BSS, such as: specifying the user-side LAN port for CPE interconnection, configuring the IP address for LAN port interconnection. The customer selects the local enterprise site and service requiring interconnection (determining the caller), then selects the remote enterprise site(s) and service(s) it intends to interconnect (determining the callee(s); multiple callees are possible). The customer selects the ordering parameters, for instant ordering, the key parameters include order duration and bandwidth. The BSS then issues a DTS-I connection setup request to the network orchestrator by calling the corresponding API.

Step 2: Resource check

Upon receiving the DTS-I order request, the Network Orchestrator first checks if both sites are available (whether the CPEs are online) and verifies if the CPEs exceed their own bandwidth limits after the order is initiated. In practical utilizations, the Network Orchestrator may select an appropriate DC based on the availability of computational resources and then establish connections according to the chosen DC. However, this process falls outside the scope of the IETF and will not be discussed here.

Step 3: VPN selection

The Network Orchestrator checks for available idle VPNs (the operator pre-configures multiple VPNs on the SGW for exclusive use by services). The Network Orchestrator selects an unused VPN for this service connection and maintains the VPN occupancy status. It then returns the order initiation result to the user (informing the user whether the order can be initiated).

Step 4: Customer-side configuration

The Network Controller configures the LAN port (specifies the IP address range of the user private network), WAN port (PPPoE over IPv6 tunnel configuration), and routing for the caller and callee site CPEs respectively.

Step 5: VPN configuration

The Network Controller configures the VPN-related parameters through SGW or the AAA system associated with the SGW.

Step 6: User authentication and authorization

After the CPE dials up, the SGW completes user account authentication and authorization. The end-to-end network connection is successfully established.

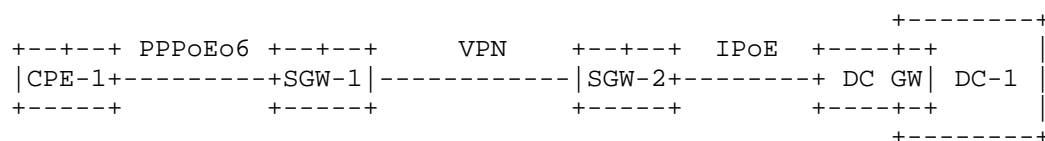


Figure 2: DTS-I Connection Example

Following the procedure above, a high-speed connection can be setup between CPE-1 and DC GW of DC-1. The whole connection consists of three segments: the segment between CPE-1 and SGW-1(based on PPPoEo6), the VPN between SGW-1 and SGW-2, and the segment between SGW-2 and DC-GW of DC-1(based on Session-level IPoE).

3.4. APIs to External Systems

The IETF has produced several YANG data models that are instrumental for automating the provisioning and delivery of connectivity services, such as such as the AC, L3SM [RFC8299], L3NM [RFC9182], L2SM [RFC8466], L2NM [RFC9291], [RFC9543] NSS YANG Model [I-D.ietf-teas-ietf-network-slice-nbi-yang], Service Attachment Points (SAPs) [RFC9408]. Network Orchestrator can use these models to set up connections between the Provider Edge devices, and also customer facing ACs between CEs and PEs, DC GWs and PEs. On the premise that these models can be directly utilized, a series of model-based APIs need be defined to meet the requirements for massive data transmission within a predetermined time period. One typical API of them is for connection setup as below,

1) Data Structure of the API Request

Parameter	Type	Meaning
MemberCount	Integer	Number of members initiating the order
CallerCompany	String	Company of the caller
CallerSite	String	Site of the caller
CallerService	String	Caller service
Callees	[Object]	Callee array
CalleeCompany	String	Company of the callee
CalleeSite	String	Site of the callee
CalleeService	String	Callee service
StartTime	String	Starting time of service
EndTime	String	End time of service
BandwidthLevel	Integer	The bandwidth level ordered: 0: 30M, 9:100M~900M, 10-19: 1G~10G

Table 1: Example of Data Structure of the API Request

2) Data Structure of the Response (with status code: 200)

Parameter	Type	Meaning
Code	Integer	Response code
Data	Object	Parameters
ConnectSessionID	String	Connection session ID after successful ordering
Message	String	Message

Table 2: Example of Data Structure of the API Response (with status code: 200)

3.5. Issues Identified

Based on the introduction above, it can be seen that the use case of DTS-I has the following requirements which may differentiate it from others,

-Ultra-high bandwidth with wide bandwidth range

The network needs to provide customers with bandwidth in various granularities, particularly ultra-high bandwidth, such as offering up to Gigabit-level bandwidth at present.

-Deterministic time frame

The DTS-I connections can be rapidly setup between the user-specified start and end points based on user requirements. The transfer of large volume of data may be finished within a predictable time frame.

- Ubiquitous Access Provisioning

The service needs ubiquitous access and wide coverage, allowing users to flexibly connect through various access methods and ensuring computational resources are readily available on demand.

-Cross-Domain Coordination

For user data transmission, the connection may span across domains or even across different operators, the network must possess cross-domain coordination capabilities, enabling flexible end-to-end scheduling of network resources and services.

The operation of such a service faces the following issues,

- Extension to the existing service models. Due to the requirements above, extension to the existing service models (e.g., L3SM) are needs to be developed for DTS-I, which can also be exposed to external systems for use through YANG2API.

- Challenges related to YANG-based API operation. Service abstractions are commonly exposed to external systems, such as orchestration platforms and OSS/BSS applications through APIs derived from YANG data models. However, the lack of consistent guidance on how abstractions should be modeled, exposed, and consumed results in APIs that vary significantly across vendors and deployments. This variability makes it difficult for external systems to consume YANG-based service APIs in a predictable and interoperable manner, operators continue to face challenges related to lifecycle management, service semantic clarity, observability, and interoperability.

- Verification of the capabilities of existing network abstractions. It is necessary to check whether the current network abstractions can support the new service model and meet the feature requirements of DTS-I; if not, the existing network abstractions will need to be updated.

In addition, the following generic issue can be identified,

Some operators are adopting TMF640/641 as APIs for service ordering from their BSS, but how these interfaces can be exposed to / aligned with the service/network models, or more broadly any YANG model, for configuration and/or state is not specified.

The declarative model-driven nature of NETCONF/YANG, and its associated semantics through NMDA, allow for the creation of complete abstractions with an unprecedented simplicity as exemplified by the LxNM and LxSM models. However, the lack of choice in commercial, off-the-shelf systems that implement such declarative methodology is seriously affecting the uptake of the protocol/modeling language, due to the risk of vendor lock-in. There are even fewer credible open-source alternatives.

An operator who offers a diverse set of customer services including L3VPN, L2VPN, (business) internet access etc. may use the LxSM models as the interface to offer these services. But the models are not well-suited to offer a combination of these, and other, services through a single service orchestrator.

The LxSM models lack administrative state control and operational state information. Additionally, in the vast majority of SP networks, configuration management and the collection of statistics / telemetry data continue to exist as two separate silos in both the organizational chart and technology stacks/APIs. This makes it very hard to bring operational state into such model.

4. Operational Challenges with Network and Service Abstractions

The previous section has demonstrated the operational issues related to this specific use case, but on a broader scale, the operation of network and service abstractions faces more challenges. While these abstractions are widely deployed, operators report persistent challenges in operationalizing them in a consistent, scalable, and automatable manner. As highlighted by the NEMOPS workshop, these challenges are systemic and operational in nature, arising from fragmented tooling, inconsistent abstraction service semantics, and limited end-to-end coordination. They are not confined to a specific technology or service type, but recur across abstraction domains and deployment environments.

4.1. Fragmented Operational Lifecycles

Operational workflows associated with service abstractions, such as service instantiation, monitoring, troubleshooting, modification, and decommissioning, are often fragmented and inconsistently handled. Even when abstractions coexist within the same network or service offering, they frequently rely on different tools, data models, and interfaces. NEMOPS discussions highlighted that operators commonly depend on a heterogeneous mix of management protocols, vendor-specific APIs, and legacy mechanisms to complete these workflows, significantly increasing operational complexity and cost.

In practice, lifecycle actions initiated through YANG-based service APIs often require coordination across orchestration systems, controllers, and device configurations. However, these components are rarely aligned in terms of lifecycle semantics, data models, or operational assumptions. This fragmentation limits end-to-end automation, complicates validation and rollback, and increases the likelihood of configuration drift and operational errors. Existing service and network abstractions generally lack native constructs to express lifecycle attributes such as activation time, duration, expiration, or rollback behavior. As a result, transient service intents must be tracked and enforced outside the abstraction framework, increasing operational complexity and the risk of inconsistency.

4.2. Misalignment Between Abstraction Layers

Service abstractions are typically realized through a combination of service-level models, network-level models, control-plane behavior, and management interfaces. These layers are often developed independently, with limited coordination across working groups or operational domains.

This misalignment can manifest as:

- Service abstractions that do not cleanly map to underlying network capabilities.
- Network models that expose parameters without clear service-level semantics.
- Control-plane behaviors that are difficult to correlate with service-level intent.

As a result, it is difficult to combine the different services into a higher level service, operators face challenges ensuring that a service behaves as intended throughout its lifecycle, particularly when changes occur at one layer without corresponding visibility or coordination at others.

4.3. Inconsistent Semantics and Operational Assumptions

Existing abstraction models often focus on configuration or control-plane aspects without fully considering how abstractions are realized operationally across networks. Service and network abstractions frequently rely on metrics, attributes, or parameters whose semantics vary across models, implementations, or consumption contexts. Concepts such as cost, availability, or performance may be represented using different definitions, units, scopes, or update models.

Many abstraction models expose parameters or metrics that are syntactically similar but semantically inconsistent across technologies or implementations. Differences in interpretation, update frequency, or scope can lead to unpredictable behavior when abstractions are consumed by automation systems. These inconsistencies complicate integration between systems and undermine the reliability of automation. These gaps are typically addressed through custom logic or manual processes, reducing portability and interoperability.

Without consistent operational semantics, it is difficult for management and orchestration systems to reliably interpret abstraction state, compare information across domains, or make automated decisions based on abstraction models alone.

4.4. Limited Feedback and Observability for Abstraction Enforcement

Existing abstractions primarily focus on configuration and offer limited standardized mechanisms for reporting whether requested behaviors have been successfully applied or remain valid over time. This lack of feedback assurance impedes closed-loop automation and increases reliance on manual monitoring and intervention.

4.5. Impact on Operational Efficiency and Interoperability

The challenges described above directly impact operational efficiency, automation, and interoperability. Operators are required to invest significant effort in integration, validation, and troubleshooting, reducing the benefits that abstractions are intended to provide. Without a more coordinated approach to abstraction modeling and operational usage, these issues are likely to persist as networks continue to evolve.

5. Operational Evidence from the IAB NEMOPS Workshop

The operational challenges described in this document are consistent with, and reinforced by, the findings of [NEMOPS] workshop, which examined the state of network management and automation based on operator experience across diverse deployment environments.

The NEMOPS workshop identified that, despite significant progress in protocol development and data modeling, operational workflows remain fragmented and difficult to automate end-to-end. Operators reported continued reliance on a heterogeneous mix of tools, protocols, and interfaces, including YANG-based management protocols, vendor-specific APIs, legacy mechanisms such as CLI and SNMP, and bespoke orchestration logic to deploy and operate services. This fragmentation increases operational complexity and limits the effectiveness of abstraction-driven automation.

A key observation from the workshop is that model-driven network management is generally successful, yet insufficient on its own to address higher-level operational needs. In particular, the workshop highlighted gaps between device-level and service-level abstractions, noting that existing models often lack the semantic alignment and contextual information required by orchestration and OSS/BSS systems. As a result, operators must perform extensive model mapping, data transformation, and system-specific integration outside the scope of standardized abstractions.

The workshop further emphasized challenges related to observability, verification, and feedback. While configuration mechanisms are relatively mature, operators reported limited ability to validate whether service intent is being met over time or to correlate operational state across abstraction layers. This lack of consistent feedback undermines closed-loop automation and complicates troubleshooting, particularly in multi-vendor and multi-domain environments.

Another recurring theme from the NEMOPS discussions is the lack of architectural documentation and operational guidance explaining how existing abstractions, models, protocols, and tools are intended to work together as a system. Operators expressed difficulty understanding which abstractions to use, how they should be combined, and how responsibilities are divided across layers and working groups. This absence of cohesive guidance leads to divergent interpretations and inconsistent deployments.

These findings closely align with the limitations identified in the applicability studies discussed earlier and reinforce a broader operational problem: while many of the necessary technical components for service and network abstractions exist, they are not sufficiently coordinated, aligned, or documented to support consistent, interoperable, and automatable operations. Addressing these systemic issues requires a focus on abstraction coherence, lifecycle semantics, and operational consumption concerns that fall squarely within the scope of the ONSEN Working Group.

6. Operational Needs Highlighted by the Use Cases

From an operational perspective, the network operation system needs to dynamically coordinate behavior across multiple network segments, expose the YANG-based network and service capabilities through open APIs, driven by service-level events, workload changes, or short-lived operational needs.

Service and network abstractions are defined and evolved across multiple IETF working groups, each focusing on a specific technology, protocol, or layer. Although this separation is appropriate for protocol development, it has resulted in abstraction models and operational assumptions that are not well coordinated across domains. As a result, operators must integrate abstractions that were designed with different scopes, semantics, and lifecycle assumptions. This fragmentation increases integration effort and complicates automation, particularly when a service abstraction spans multiple technologies or administrative domains.

YANG data models are commonly used as the basis for APIs that expose service abstractions to external systems. However, existing work provides limited guidance on how these abstractions should be exposed, versioned, or consumed in a predictable and interoperable manner. As a result, APIs derived from similar abstraction models may differ significantly across vendors or deployments, requiring bespoke integration by operators and OSS/BSS systems. Some operators are adopting TMF640/641 as APIs for service ordering from their BSS, but how these interfaces can be exposed to / aligned with the service/network models, or more broadly any YANG model, for configuration and/or state is not specified. This variability undermines the portability and reuse that abstractions are intended to provide.

To address the issues above, a new Working Group is needed to perform the following tasks:

- Identify these gaps and provide guidance and recommendations on how YANG-based service APIs should express and expose such behaviors to better support dynamic, multi-operator environments.
- Maintaining YANG data models for network and service abstractions.
- Evaluating whether YANG data model activities above necessitate changing the Automating Service and Network Management Framework defined in [RFC8969]
- Provide YANG-based API tooling-related guidances and document in WG-maintained repositories such as GitHub or a Wiki.

7. Operational Considerations

TBD.

8. Security Considerations

TBD.

9. IANA Considerations

No Action is needed.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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>>.
- [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>>.
- [RFC8969] Wu, Q., Ed., Boucadair, M., Ed., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <<https://www.rfc-editor.org/info/rfc8969>>.
- [RFC9182] Barguil, S., Gonzalez de Dios, O., Ed., Boucadair, M., Ed., Munoz, L., and A. Aguado, "A YANG Network Data Model for Layer 3 VPNs", RFC 9182, DOI 10.17487/RFC9182, February 2022, <<https://www.rfc-editor.org/info/rfc9182>>.
- [RFC9291] Boucadair, M., Ed., Gonzalez de Dios, O., Ed., Barguil, S., and L. Munoz, "A YANG Network Data Model for Layer 2 VPNs", RFC 9291, DOI 10.17487/RFC9291, September 2022, <<https://www.rfc-editor.org/info/rfc9291>>.

- [RFC9543] Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "A Framework for Network Slices in Networks Built from IETF Technologies", RFC 9543, DOI 10.17487/RFC9543, March 2024, <<https://www.rfc-editor.org/info/rfc9543>>.

10.2. Informative References

- [I-D.bg-onions-update-network-service-models]
de Dios, O. G. and S. Barguil, "An Update of Service and Network YANG Data Models", Work in Progress, Internet-Draft, draft-bg-onions-update-network-service-models-00, 16 September 2025, <<https://datatracker.ietf.org/doc/html/draft-bg-onions-update-network-service-models-00>>.
- [I-D.dunbar-neotec-ac-pe2pe-ucmp-applicability]
Dunbar, L., Qiong, S., Wu, B., Contreras, L. M., and C. Xie, "Applying Attachmet Circuit and PE2PE YANG Data Model to dynamic policies Use Case", Work in Progress, Internet-Draft, draft-dunbar-neotec-ac-pe2pe-ucmp-applicability-01, 22 June 2025, <<https://datatracker.ietf.org/doc/html/draft-dunbar-neotec-ac-pe2pe-ucmp-applicability-01>>.
- [I-D.dunbar-onions-ac-te-applicability]
Dunbar, L., Qiong, S., Wu, B., Contreras, L. M., and C. Xie, "Applying Attachmet Circuit and Traffic Engineering YANG Data Model to Edge AI Use Case", Work in Progress, Internet-Draft, draft-dunbar-onions-ac-te-applicability-00, 3 October 2025, <<https://datatracker.ietf.org/doc/html/draft-dunbar-onions-ac-te-applicability-00>>.
- [I-D.ietf-teas-ietf-network-slice-nbi-yang]
Wu, B., Dhody, D., Rokui, R., Saad, T., and J. Mullooly, "A YANG Data Model for the RFC 9543 Network Slice Service", Work in Progress, Internet-Draft, draft-ietf-teas-ietf-network-slice-nbi-yang-25, 9 May 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-ietf-network-slice-nbi-yang-25>>.
- [NEMOPS] "NEMOPS", <<https://datatracker.ietf.org/group/nemopsws/about/>>.
- [RFC8975] Kuhn, N., Ed. and E. Lochin, Ed., "Network Coding for Satellite Systems", RFC 8975, DOI 10.17487/RFC8975, January 2021, <<https://www.rfc-editor.org/info/rfc8975>>.

- [RFC9408] Boucadair, M., Ed., Gonzalez de Dios, O., Barguil, S., Wu, Q., and V. Lopez, "A YANG Network Data Model for Service Attachment Points (SAPs)", RFC 9408, DOI 10.17487/RFC9408, June 2023, <<https://www.rfc-editor.org/info/rfc9408>>.
- [RFC9834] Boucadair, M., Ed., Roberts, R., Ed., Gonzalez de Dios, O., Barguil, S., and B. Wu, "YANG Data Models for Bearers and Attachment Circuits as a Service (ACaaS)", RFC 9834, DOI 10.17487/RFC9834, September 2025, <<https://www.rfc-editor.org/info/rfc9834>>.

Authors' Addresses

Chongfeng Xie
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China
Email: xiechf@chinatelecom.cn

Qiong Sun
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China
Email: sunqiong@chinatelecom.cn

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

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Madrid
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

Kris Lambrechts
Netedge
Email: kris@netedge.plus

Bo Wu
Huawei
Email: lana.wubo@huawei.com