Network Management Rearch Group
Internet-Draft
Intended status: Informational
Expires: 4 May 2021

D. Bogdanovic
X. Liu
Volta Networks
L.M. Contreras
Telefonica

31 October 2020

Multilevel configuration
draft-bogdanovic-multilevel-configuration-00

Abstract

This document describes issues caused by residual configurations in network devices and how multi-level configuration could potentially offer a solution.

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 4 May 2021.

Table of Contents

1. Definitions and Acronyms

TCAM: Ternary Content Addressable Memory

2. Introduction

As network operators experience traffic and customer growth, the network device configurations are getting larger. All the config information, both network operator and customers, on the device is multiplexed into single file and the configuration differentiation belonging to different owners becomes harder. This leads to the operators not knowing why certain parts of the config are in the file. Another issue contributing to config growth are debugging sessions. Network operator enters the device and starts editing configuration. After the debug session is finnished, it is not unusual for debug configuration entries to stay in the config file indefinitely.

In order to solve this problem, some operators created central database with all the network configuration files that act as systems of record. If anything is to persist on the device in the network, it has to be in the central database. Still, this solution has not remedided the problem.

Both, vendors and operators, contribute to the problem:

* Vendors by keeping the configuration file structures as currently designed;

* Operators by allowing human operator to directly edit config file on the device.

Until the above two issues are solved, the residual configuration problem will persist and continue to waste expensive data plane resources (TCAM).

This draft authors are motivated to propose a solution from both sides, operator and vendor. Our initial idea is to keep the persistent configuration at minimum on the device. All network service configurations are generated on demand and are ephemeral. This requires a change to the config file structure, creating multi-level file structure with dependencies between different levels.Besides the residual configuration problem, there are other use cases that multi-level configuration can be applied, that are listed in this document.

3. Use cases

3.1. Service assurance

Service assurance is one of the critical operational aspects of the communication networks. As [I-D.claise-opsawg-service-assurance-architecture] states, services rely on multiple sub-services on top of the same underlying network, then service affection on any of those sub-services can propagate impacts to many other services in the network. In this respect, the multi-level network configuration approach could help on identifying by design the correlation among services and atomic functions in the network, simplifying the operation and providing a uniform framework across networks.

3.2. Network migrations and mergings

Quite often service providers get involved in complex procedures of network mergings or migrations. Either driven by simplification of existing networks, introduction of new services, rationalization of multiple infrastructures, acquisition of other providers, etc., all of them imply both the introduction and removal of distinct configurations of multiple purposes. Apart of the complexity and difficulty of converging to a common and unique approach, these procedures could impact service continuity. In this sense, multi-level network configuration could highly simplify the process. First, by dividing the problem in smaller pieces, dealing with the issue per configuration level instead of considering the whole configuration. And second, by allowing incremental execution of the process by acting on particular levels each time.

3.3. Network slicing

Network slices are expected to provide tailored networks that can accommodate services with specific characteristics and service level objectives (SLOs) [I-D.nsdt-teas-ietf-network-slice-definition]. In this respect, the multi-level network configuration approach can be leveraged as a mean for deploying particular IETF network slices, facilitating the instantiation, operation and decommissioning of the slice in a straightforward manner.

3.4. Zero touch provisioning

[RFC8886] proposes a mechanism for remotely auto-installing configurations on network devices with proper confidentiality and security. Such mechanism is conceived for receiving initial configuration by the device, for a later completion of the configuration by other means. In this case, leveraging on multi-level network configuration could permit incremental deployment of configuration levels following a similar auto-installing approach, according to some configuration workflow as defined by the service provider.

4. Security Considerations

TBD

5. IANA Considerations

This document currently has no items for IANA considerations.

6. Acknowledgements

7. Change log [RFC Editor: Please remove]

8. Informative References

[I-D.claise-opsawg-service-assurance-architecture]
Claise, B., Quilbeuf, J., Fathi, Y., Lopez, D., and D. Voyer, "Service Assurance for Intent-based Networking Architecture", Work in Progress, Internet-Draft, draft-claise-opsawg-service-assurance-architecture-03, 27 July 2020, <http://www.ietf.org/internet-drafts/draft-claise-opsawg-service-assurance-architecture-03.txt>.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", Work in Progress, Internet-Draft, draft-nsdt-teas-ietf-network-slice-definition-00, 21 October 2020, <http://www.ietf.org/internet-drafts/draft-nsdt-teas-ietf-network-slice-definition-00.txt>.

[RFC8886] Kumari, W. and C. Doyle, "Secure Device Install", RFC 8886, DOI 10.17487/RFC8886, September 2020, <https://www.rfc-editor.org/info/rfc8886>.

Authors' Addresses

Dean Bogdanovic
Volta Networks

Email: dean@voltanet.io

Xufeng Liu
Volta Networks

Email: xufeng@voltant.io

Luis M. Contreras
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com

NMRG
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

LM. Contreras
Telefonica
P. Demestichas
WINGS
J. Tantsura
Apstra, Inc.

November 2, 2020

IETF Network Slice Intent
draft-contreras-nmrg-transport-slice-intent-04

Abstract

Slicing at the transport network is expected to be offered as part of end-to-end network slices, fostered by the introduction of new services such as 5G. This document explores the usage of intent technologies for requesting IETF network slices.

Table of Contents

1. Introduction

Network slicing is emerging as the future model for service offering in telecom operator networks. Conceptually, network slicing provides a customer with an apparent dedicated network built on top of logical (i.e. virtual) and/or physical functions and resources supported by a shared infrastructure, provided by one or more telecom operators.

The concept of network slicing has been largely fostered by the advent of 5G services that are expected to be deployed on top of different kind of slices, each built to support specific characteristics (extreme low latency, high bandwidth, etc).

As part of an end-to-end network slice it is expected to have a number of network slices at transport level (referred as IETF network slices) providing the necessary connectivity to the rest of components of the end-to-end slice, e.g., mobile packet core slice.

For a definition of an IETF network slice refer to [I-D.nsdt-teas-ietf-network-slice-definition]. The following paragraph is directly taken from it: "An IETF Network Slice is a logical network topology connecting a number of endpoints with a set of shared or dedicated network resources, that are used to satisfy specific Service Level Objectives (SLOs)."

Intent is a high-level, declarative goal that operates at the level of a network and services it provides, not individual devices. It is used to define outcomes and high-level operational goals.

In consequence, it seems very convenient to apply the intent-based mechanisms for the provision of IETF network slices, providing the adequate level of abstraction towards the transport network control and management planes.

This document leverages current industry trends in the definition of end-to-end network slices. The final objective is to describe intents that can be used to flexibly declare the operational aspects and goals of an IETF network slice, meaning that the customer could declare what kind of IETF network slice is needed (the outcome) and not how to achieve the goals of the IETF network slice.

2. IETF network slice intent

As stated in [I-D.irtf-nmrg-ibn-concepts-definitions], "Intent is a declaration of operational goals that a network should meet and outcomes that the network is supposed to deliver, without specifying how to achieve them. Those goals and outcomes are defined in a manner that is purely declarative - they specify what to accomplish, not how to achieve it."

When applied to transport networks, this implies that an intent for IETF network slices should provide the necessary abstraction with respect to implementation details, including the final devices (or resources) involved, and be focused on the characteristics and performance expectations related to it.

With that intent it can be expected that the intent based system can fulfill and assure the requested IETF network slice, triggering initial configurations at the time of initial provisioning and corrective actions during the IETF network slice lifetime.

3. Foundation of IETF network slice intents

The industrial interest around 5G is accelerating network deployments and operational changes.

With this respect, the GSMA has been developing a universal blueprint that can be used by any vertical customer to request the deployment of a network slice instance (NSI) based on a specific set of service requirements. Such a blueprint is a network slice descriptor called Generic Slice Template (GST) [GSMA]. The GST contains multiple attributes that can be used to characterize a network slice. A particular template filled with values generates a specific Network Slice Type(NEST).

Such templates refer to the end-to-end network slice, including the transport part. Despite the fact that some of the values would not

have applicability for the transport network, others do. An analysis of the relevant attributes is performed in [I-D.contreras-teas-slice-nbi].

According to 3GPP propositions [TS28.541], an upper 3GPP Management System interacts with the transport network for establishing the necessary slices at the transport level. Such interaction can be expected to happen using the IETF network slice intent, described to an intent-based system (IBS) in the transport network part. Then, according to the intent lifecycle in [I-D.irtf-nmrg-ibn-concepts-definitions], the IBS, after recognizing the intent, will proceed to translate it in order to interact with a IETF network slice controller by using a NBI as proposed in [I-D.contreras-teas-slice-nbi].

4. Mechanisms for translating IETF network slice intents

This section describes approaches for implementing mechanisms to translate IETF network slice intents.

4.1. Translation approaches and interaction with the upper systems

A suite of mechanisms will be required to allow instantiation of the user's intent into a IETF network slice. In order to be able to deliver an end2end Intent driven slice - a well defined set of context aware attributes that allow unambiguous instantiation of the intent should be agreed upon. A combination of a structured set of attributes communicated between an IBN and an upper layer system with user input would allow an IBN to have intent modeled and reason about its completeness/validity. Translation approaches and interaction with the upper systems might benefit from Natural Language Processing (NLP) technics that are needed for enabling high level expression of requirements found missing. The goal would be to identify and classify the answers for as many fields as possible from the Generic Slice Template (GST), based on the free text / speech provided by the user. As it is highly unlikely that the minimum set of fields to properly define an IETF network slice (geo-temporal characteristics, performance characteristics, SLO and SLA properties) will be fulfilled in this first step, a follow up two-step approach might need to be implemented.

- The minimum missing fields from the GST have to be identified and appropriate questions have to be generated (e.g. based on a pool of available questions correlated with each field, or based on AI approaches).

- An iterative interrogation phase will be initiated towards the user using the previously generated questions, until the user

provides all the missing information, so the intent can be modeled accordingly.

Interaction with the user and higher-up systems can potentially be further improved by utilizing Machine Learning techniques.

4.2. Intent-based system suite

In order to consolidate on the set of devices, technologies and resources to be used, a combination of deterministic or stochastic computation approaches will be needed. Deterministic approaches will rely on mathematical models and respective algorithms. Stochastic approaches will rely on technologies like machine learning. Their goal will be to learn from experience, so as to optimize future decisions from the viewpoint of speed and reliability. The target of learning will be related to the service behavior and to the anticipated network status in the area and time period of the service provision.

5. Security Considerations

To be done.

6. IANA Considerations

This draft does not include any IANA considerations

7. References

[GSMA] "Generic Network Slice Template, version 3.0", NG.116 , May 2020.

[I-D.contreras-teas-slice-nbi] Contreras, L., Homma, S., and J. Ordonez-Lucena, "IETF Network Slice use cases and attributes for Northbound Interface of controller", draft-contreras-teas-slice-nbi-03 (work in progress), October 2020.

[I-D.irtf-nmrg-ibn-concepts-definitions] Clemm, A., Ciavaglia, L., Granville, L., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", draft-irtf-nmrg-ibn-concepts-definitions-02 (work in progress), September 2020.

[I-D.nsdt-teas-ietf-network-slice-definition]
Rokui, R., Homma, S., Makhijani, K., Contreras, L., and J. Tantsura, "Definition of IETF Network Slices", draft-nsdt-teas-ietf-network-slice-definition-00 (work in progress), October 2020.

[TS28.541]
"TS 28.541 Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16) V16.2.0.", 3GPP TS 28.541 V16.2.0 , September 2019.

Acknowledgments

This work has been partly funded by the European Commission through the H2020 project 5G-EVE (Grant Agreement no. 815074).

Contributors

Kostas Tsagkaris, Kostas Trichias, Vassilis Foteinos, and Thanasis Gkiolias (all from WINGS ICT Solutions) have also contributed to this work.

Authors' Addresses

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com/

Panagiotis Demestichas
WINGS ICT Solutions
Greece

Email: pdemest@wings-ict-solutions.eu

Jeff Tantsura
Apstra, Inc.

Email: jefftant.ietf@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 19, 2021

A. Clemm
Futurewei
L. Ciavaglia
Nokia
L. Granville
Federal University of Rio Grande do Sul (UFRGS)
J. Tantsura
Apstra, Inc.

September 15, 2020

Intent-Based Networking - Concepts and Definitions
draft-irtf-nmrg-ibn-concepts-definitions-02

Abstract

Intent and Intent-Based Networking (IBN) are taking the industry by storm. At the same time, those terms are used loosely and often inconsistently, in many cases overlapping and confused with other concepts such as "Policy". This document clarifies the concept of "Intent" and provides an overview of functionality that is associated with it. The goal is to contribute towards a common and shared understanding of terms, concepts, and functionality that can be used as foundation to guide further definition of associated research and engineering problems and their solutions.

Table of Contents

1. Introduction

Traditionally in the IETF, interest regarding management and operations has focused on individual network and device features. Standardization emphasis has generally been put on management instrumentation that needed to be provided to a networking device. A

prime example of this is SNMP-based management and the 200+ MIBs that have been defined by the IETF over the years. More recent examples include YANG data model definitions for aspects such as interface configuration, ACL configuration, or Syslog configuration.

There is a sense and reality that in modern network environments managing networks by configuring myriads of "nerd knobs" on a device-by-device basis is no longer sustainable. Significant challenges arise with keeping device configurations not only consistent across a network, but consistent with the needs of services and service features they are supposed to enable. Adaptability to changes at scale is a fundamental property of a well-designed IBN system, that requires the ability to consume and process analytics that is context/intent aware at near real-time speeds. At the same time, operations need to be streamlined and automated wherever possible to not only lower operational expenses, but also allow for rapid reconfiguration of networks at sub-second time scales and to ensure that networks are delivering their functionality as expected.

Accordingly, the IETF has begun to address end-to-end management aspects that go beyond the realm of individual devices in isolation. Examples include the definition of YANG models for network topology [RFC8345] or the introduction of service models used by service orchestration systems and controllers [RFC8309]. Much interest has been fueled by the discussion about how to manage autonomic networks, as discussed in the ANIMA working group. Autonomic networks are driven by the desire to lower operational expenses and make the management of the network as a whole more straightforward, putting it at odds with the need to manage the network one device and one feature at a time. However, while autonomic networks are intended to exhibit "self-management" properties, they still require input from an operator or outside system to provide operational guidance and information about the goals, purposes, and service instances that the network is to serve.

This vision has since caught on with the industry in a big way, leading to a significant number of solutions that offer "Intent-based management" that promise network providers to manage networks holistically at a higher level of abstraction and as a system that happens to consist of interconnected components, as opposed to a set of independent devices (that happen to be interconnected). Those offerings include IBN systems (offering full a life-cycle of intent), SDN controllers (offering a single point of control and administration for a network), and network management and Operations Support Systems (OSS).

However, it has been recognized for a long time that comprehensive management solutions cannot operate only at the level of individual

devices and low-level configurations. In this sense, the vision of "Intent" is not entirely new. In the past, ITU-T's model of a Telecommunications Management Network, TMN, introduced a set of management layers that defined a management hierarchy, consisting of network element, network, service, and business management. High-level operational objectives would propagate in a top-down fashion from upper to lower layers. The associated abstraction hierarchy was crucial to decompose management complexity into separate areas of concerns. This abstraction hierarchy was accompanied by an information hierarchy that concerned itself at the lowest level with device-specific information, but that would, at higher layers, include, for example, end-to-end service instances. Similarly, the concept of "Policy-based Network Management (PBNM)" has, for a long time, touted the ability to allow users to manage networks by specifying high-level management policies, with policy systems automatically "rendering" those policies, i.e., breaking them down into low-level configurations and control logic.

What has been missing, however, is putting these concepts into a more current context and updating them to account for current technology trends. This document clarifies the concepts behind intent. It differentiates it from related concepts. It also provides an overview of first-order principles of Intent-Based Networking as well as associated functionality. The goal is to contribute to a common and shared understanding that can be used as a foundation to articulate research and engineering problems in the area of Intent-Based Networking. It should be noted that the articulation of those problems is beyond this document's scope.

2. Key Words

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.

3. Definitions and Acronyms

ACL: Access Control List

API: Application Programming Interface

Intent: A set of operational goals that a network should meet and outcomes that a network is supposed to deliver, defined in a declarative manner without specifying how to achieve or implement them.

IBA: Intent-Based Analytics - Analytics that are defined and derived from users' intent and used to validate the intended state.

IBN: Intent-Based Network, a network that can be managed using intent.

IBS: Intent-Based System, a system that supports management functions that can be guided using intent.

Policy: A set of rules that governs the choices in behavior of a system.

PDP: Policy Decision Point

PEP: Policy Enforcement Point

Service Model: A model that represents a service that is provided by a network to a user.

SSoT: Single Source of Truth - A functional block in an IBN system that normalizes users' intent and serves as the single source of data for the lower layers.

4. Introduction of Concepts

The following section provides an overview of the concept of Intent and Intent-Based Management. It also provides an overview of the related concepts of service models, and of policies respectively Policy-Based Network Management, and explains how they relate to Intent and Intent-Based Management.

4.1. Intent and Intent-Based Management

The term "Intent" was first introduced in the context of Autonomic Networks, where it is defined as "an abstract, high-level policy used to operate a network" [RFC7575]. According to this definition, an Intent is a specific type of policy, provided by a user to provide guidance to the Autonomic Network that would otherwise operate without human intervention. However, to avoid using "Intent" simply as a synonym for "Policy", a distinction needs to be introduced that differentiates Intent clearly from other types of policies.

For one, while Intent-Based Management aims to lead towards networks that are dramatically simpler to manage and operate requiring only minimal outside intervention, the concept of "Intent" is not limited to autonomic networks, but applies to any network. Networks, even when considered "autonomic", are not clairvoyant and have no way of

automatically knowing particular operational goals nor what instances of networking services to support. In other words, they do not know what the "Intent" of the network provider is that gives the network the purpose of its being. This still needs to be communicated by what informally constitutes "Intent".

More specifically, Intent is a declaration of operational goals that a network should meet and outcomes that the network is supposed to deliver, without specifying how to achieve them. Those goals and outcomes are defined in a manner that is purely declarative - they specify what to accomplish, not how to achieve it. "Intent" thus applies several important concepts simultaneously:

- It provides data abstraction: Users and operators do not need to be concerned with low-level device configuration and nerd knobs.

- It provides functional abstraction from particular management and control logic: Users and operators do not need to be concerned even with how to achieve a given Intent. What is specified is a desired outcome, with the Intent-based system automatically figuring out a course of action (e.g., a set of rules (thus, a set of rules is not part of an intent but rather derived from that intent), an algorithm) for how to achieve the outcome.

The following are some examples of intent:

- "Steer networking traffic originating from endpoints in one geography away from a second geography, unless the destination lies in that second geography."

- "Avoid routing networking traffic originating from a given set of endpoints (or associated with a given customer) through a particular vendor's equipment, even if this occurs at the expense of reduced service levels."

- "Maximize network utilization even if it means trading off service levels (such as latency, loss), unless service levels have deteriorated 20% or more from their historic mean."

- "VPN service must have path protection at all times for all paths."

- "Generate in-situ OAM data and network telemetry across for later offline analysis whenever significant fluctuations in latency across a path are observed."

In an autonomic network, intent should be rendered by the network itself, i.e., translated into device-specific rules and courses of

action. Ideally, it should not even be orchestrated or broken down by a higher-level, centralized system, but by the network devices themselves using a combination of distributed algorithms and local device abstraction. In this idealized vision, because intent holds for the network as a whole, intent should ideally be automatically disseminated across all devices in the network, which can themselves decide whether they need to act on it.

However, such decentralization will not be practical in all cases. Certain functions will need to be at least conceptually centralized. For example, users may require a single conceptual point of interaction with the network. Likewise, the vast majority of network devices may be intent-agnostic and focus only (for example) on the actual forwarding of packets. This implies that certain intent functionality needs to be provided by functions that are specialized for that purpose (which depending on scenario may be hosted on dedicated systems, or cohosted with other networking functions). For example, functionality to translate intent into courses of actions and algorithms to achieve desired outcomes may need to be provided by such specialized functions. Of course, to avoid single points of failure, the implementation and hosting of those functions may still itself be distributed, even if conceptually centralized.

Accordingly, an intent-based network is a network that can be managed using intent. This means, it is able to recognize and ingest intent of an operator, or user, and configure and adapt itself autonomously according to the user intent, achieving an intended outcome (i.e., a desired state or behavior) without requiring the user to specify the detailed technical steps for how to achieve the outcome. Instead, the intent-based network will be able to figure out on its own how to achieve the outcome.

Other definitions of intent exist, such as [TR523]. Intent there is simply defined as a declarative interface that is typically provided by a controller. It implies the presence of a centralized function that renders the intent into lower-level policies or instructions and orchestrates them across the network. While this is certainly one way of implementation, the definition presented here is narrower in the sense that it emphasizes the importance of managing the network by specifying desired outcomes without the specific steps to be taken in order to achieve the outcome. A controller API that simply provides a network-level of abstraction would not necessarily qualify as intent. Likewise, ingestion and recognition of intent may not necessarily occur via a traditional API, but may involve other types of human-machine interactions.

4.2. Related Concepts

4.2.1. Service Models

A service model is a model that represents a service that is provided by a network to a user. Per [RFC8309], a service model describes a service and its parameters in a portable/implementation-agnostic way that can be used independently of the equipment and operating environment on which the service is realized. Two subcategories are distinguished: a "Customer Service Model" describes an instance of a service as provided to a customer, possibly associated with a service order. A "Service Delivery Model" describes how a service is instantiated over existing networking infrastructure.

An example of a service could be a Layer 3 VPN service [RFC8299], a Network Slice, or residential Internet access. Service models represent service instances as entities in their own right. Services have their own parameters, actions, and life-cycles. Typically, service instances can be bound to end-users, who might be billed for the service.

Instantiating a service typically involves multiple aspects:

- A user (or northbound system) needs to define and/or request a service to be instantiated.

- Resources need to be allocated, such as IP addresses, AS numbers, VLAN or VxLAN pools, interfaces, bandwidth, or memory.

- How to map services to the resources needs to be defined. Multiple mappings are often possible, which to select may depend on context (such as which type of access is available to connect the end user with the service).

- Bindings need to be maintained between upper and lower-level objects.

- Once instantiated, the service needs to be validated and assured to ensure that the network indeed delivers the service as requested.

They involve a system, such as a controller, that provides provisioning logic. This includes breaking down high-level abstractions into lower-level device abstractions, identifying and allocating system resources, and orchestrating individual provisioning steps. Orchestration operations are generally conducted using a "push" model in which the controller/manager initiates the operations as required, then pushes down the specific configurations

to the device. In addition to instantiating and creating new instances of a service, updating, modifying, and decommissioning services need to be also supported. The device itself typically remains agnostic to the service or the fact that its resources or configurations are part of a service/concept at a higher layer.

Instantiated service models map to instantiated lower-layer network and device models. Examples include instances of paths, or instances of specific port configurations. The service model typically also models dependencies and layering of services over lower-layer networking resources that are used to provide services. This facilitates management by allowing to follow dependencies for troubleshooting activities, to perform impact analysis in which events in the network are assessed regarding their impact on services and customers. Services are typically orchestrated and provisioned top-to-bottom, which also facilitates keeping track of the assignment of network resources. Service models might also be associated with other data that does not concern the network but provides business context. This includes things such as customer data (such as billing information), service orders and service catalogs, tariffs, service contracts, and Service Level Agreements (SLAs), including contractual agreements regarding remediation actions.

[I-D.ietf-teas-te-service-mapping-yang] is an example of a data model that provides a mapping for customer service models (e.g., the L3VPN Service Model) to Traffic Engineering (TE) models (e.g., the TE Tunnel or the Abstraction and Control of Traffic Engineered Networks Virtual Network model)

Like intent, service models provide higher layers of abstraction. Service models are often also complemented with mappings that capture dependencies between service and device or network configurations. Unlike intent, service models do not allow to define a desired "outcome" that would be automatically maintained by the intent system. Instead, the management of service models requires the development of sophisticated algorithms and control logic by network providers or system integrators.

4.2.2. Policy and Policy-Based Network Management

Policy-Based Network Management (PBNM) is a management paradigm that separates the rules that govern the behavior of a system from the functionality of the system. It promises to reduce maintenance costs of information and communication systems while improving flexibility and runtime adaptability. It is present today at the heart of a multitude of management architectures and paradigms, including SLA-driven, Business-driven, autonomous, adaptive, and self-* management [Boutaba07]. The interested reader is asked to refer to the rich set

of existing literature, which includes this and many other references. In the following, we will only provide a much-abridged and distilled overview.

At the heart of policy-based management is the concept of a policy. Multiple definitions of policy exist: "Policies are rules governing the choices in the behavior of a system" [Sloman94]. "Policy is a set of rules that are used to manage and control the changing and/or maintaining of the state of one or more managed objects" [Strassner03]. Common to most definitions is the definition of a policy as a "rule". Typically, the definition of a rule consists of an event (whose occurrence triggers a rule), a set of conditions (which get assessed and which must be true before any actions are actually "fired"), and finally a set of one or more actions that are carried out when the condition holds.

Policy-based management can be considered an imperative management paradigm: Policies precisely specified what needs to be done when and in which circumstance. By using policies, management can, in effect, be defined as a set of simple control loops. This makes policy-based management a suitable technology to implement autonomic behavior that can exhibit self-* management properties, including self-configuration, self-healing, self-optimization, and self-protection. In effect, policies define management as a set of simple control loops.

Policies typically involve a certain degree of abstraction in order to cope with the heterogeneity of networking devices. Rather than having a device-specific policy that defines events, conditions, and actions in terms of device-specific commands, parameters, and data models, a policy is defined at a higher-level of abstraction involving a canonical model of systems and devices to which the policy is to be applied. A policy agent on a controller or the device subsequently "renders" the policy, i.e., translates the canonical model into a device-specific representation. This concept allows applying the same policy across a wide range of devices without needing to define multiple variants. In other words - policy definition is de-coupled from policy instantiation and policy enforcement. This enables operational scale and allows network operators and authors of policies to think in higher terms of abstraction than device specifics and be able to reuse the same, high-level definition across different networking domains, WAN, DC, or public cloud.

PBNM is typically "push-based": Policies are pushed onto devices where they are rendered and enforced. The push operations are conducted by a manager or controller, which is responsible for deploying policies across the network and monitor their proper

operation. That being said, other policy architectures are possible. For example, policy-based management can also include a pull-component in which the decision regarding which action to take is delegated to a so-called Policy Decision Point (PDP). This PDP can reside outside the managed device itself and has typically global visibility and context with which to make policy decisions. Whenever a network device observes an event that is associated with a policy, but lacks the full definition of the policy or the ability to reach a conclusion regarding the expected action, it reaches out to the PDP for a decision (reached, for example, by deciding on an action based on various conditions). Subsequently, the device carries out the decision as returned by the PDP - the device "enforces" the policy and hence acts as a PEP (Policy Enforcement Point). Either way, PBNM architectures typically involve a central component from which policies are deployed across the network, and/or policy decisions served.

Like Intent, policies provide a higher layer of abstraction. Policy systems are also able to capture dynamic aspects of the system under management through the specification of rules that allow defining various triggers for specific courses of actions. Unlike intent, the definition of those rules (and courses of actions) still needs to be articulated by users. Since the intent is unknown, conflict resolution within or between policies requires interactions with a user or some kind of logic that resides outside of PBM. In that sense, policy constitutes a lower level of abstraction than intent, and it is conceivable for Intent-Based Systems to generate policies that are subsequently deployed by a PBM, allowing PBM to support Intent-Based Networking.

4.2.3. Distinguishing between Intent, Policy, and Service Models

What Intent, Policy, and Service Models all have in common is the fact that they involve a higher-layer of abstraction of a network that does not involve device-specifics, that generally transcends individual devices, and that makes the network easier to manage for applications and human users compared to having to manage the network one device at a time. Beyond that, differences emerge. Service models have less in common with policy and intent than policy and intent do with each other.

Summarized differences:

- A service model is a data model that is used to describe instances of services that are provided to customers. A service model has dependencies on lower-level models (device and network models) when describing how the service is mapped onto underlying network and IT infrastructure. Instantiating a service model requires

orchestration by a system; the logic for how to orchestrate/manage/provide the service model, and how to map it onto underlying resources, is not included as part of the model itself.

- Policy is a set of rules, typically modeled around a variation of events/conditions/actions, used to express simple control loops that can be rendered by devices, without requiring intervention by the outside system. Policy lets users define what to do under what circumstances, but it does not specify the desired outcome.

- Intent is a high-level, declarative goal that operates at the level of a network and services it provides, not individual devices. It is used to define outcomes and high-level operational goals, without specifying how those outcomes and should be achieved or how goals should specifically be satisfied, and without the need to enumerate specific events, conditions, and actions. Which algorithm or rules to apply can be automatically "learned/derived from intent" by the intent system. In the context of autonomic networking, intent is ideally rendered by the network itself; also, the dissemination of intent across the network and any required coordination between nodes is resolved by the network without the need for external systems.

One analogy to capture the difference between policy and intent systems is that of Expert Systems and Learning Systems in the field of Artificial Intelligence. Expert Systems operate on knowledge bases with rules that are supplied by users, analogous to policy systems whose polies are supplied by users. They are able to make automatic inferences based on those rules, but are not able to "learn" new rules on their own. Learning Systems (popularized by deep learning and neural networks), on the other hand, are able to learn without depending on user programming or articulation of rules. However, they do require a learning or training phase, and explanations of actions that the system actually takes provide a different set of challenges. Analogous to intent-based systems, users focus on what they would like the learning system to accomplish, but not how to do it.

5. Principles

The following main operating principles allow characterizing the intent-based/-driven/-defined nature of a system.

1. Single Source of Truth (SSoT) and Single Version/View of Truth (SVoT). The SSoT is an essential component of an intent-based system as it enables several important operations. The set of validated intent expressions is the system’s SSoT. SSoT and the

records of the operational states enable comparing the intended state and actual state of the system and determining drift between them. SSoT and the drift information provide the basis for corrective actions. If the intent-based is equipped with prediction capabilities or means, it can further develop strategies to anticipate, plan, and pro-actively act on the diverging trends with the aim to minimize their impact. Beyond providing a means for consistent system operation, SSoT also allows for better traceability to validate if/how the initial intent and associated business goals have been properly met, to evaluate the impacts of changes in the intent parameters and impacts and effects of the events occurring in the system. Single Version (or View) of Truth derives from the SSoT and can be used to perform other operations such as query, poll, or filter the measured and correlated information to create so-called "views". These views can serve the operators and/or the users of the intent-based system. To create intents as single sources of truth, the intent-based system must follow well-specified and well-documented processes and models. In other contexts [Lenrow15], SSoT is also referred to as the invariance of the intent.

2. One-touch but not one-shot. In an ideal intent-based system, the user expresses its intents in one form or another, and then the system takes over all subsequent operations (one-touch). A zero-touch approach could also be imagined in the case where the intent-based system has the capabilities or means to recognize intentions in any form of data. However, the zero- or one-touch approach should not be mistaken the fact that reaching the state of a well-formed and valid intent expression is not a one-shot process. On the contrary, the interfacing between the user and the intent-based system could be designed as an interactive and iterative process. Depending on the level of abstraction, the intent expressions will initially contain more or less implicit parts, and unprecise or unknown parameters and constraints. The role of the intent-based system is to parse, understand, and refine the intent expression to reach a well-formed and valid intent expression that can be further used by the system for the fulfillment and assurance operations. An intent refinement process could use a combination of iterative steps involving the user to validate the proposed refined intent and to ask the user for clarifications in case some parameters or variables could not be deduced or learned by the means of the system itself. In addition, the Intent-Based System will need to moderate between conflicting intent, helping users to properly choose between intent alternatives that may have different ramifications.

3. Autonomy and Supervision. A desirable goal for an intent-based system is to offer a high degree of flexibility and freedom on both the user side and system side, e.g., by giving the user the ability to express intents using its own terms, by supporting different forms of expression of intents and being capable of refining the intent expressions to well-formed and exploitable expressions. The dual principle of autonomy and supervision allows to operate a system that will have the necessary levels of autonomy to conduct its tasks and operations without requiring intervention of the user and taking its own decisions (within its areas of concern and span of control) as how to perform and meet the user expectations in terms of performance and quality, while at the same time providing the proper level of supervision to satisfy the user requirements for reporting and escalation of relevant information.

4. Learning. An intent-based system is a learning system. By contrast to the imperative type of system, such as Event-Condition-Action policy rules, where the user defines beforehand the expected behavior of the system to various events and conditions, in an intent-based system, the user only declares what the system should achieve and not how to achieve these goals. There is thus a transfer of reasoning/rationality from the human (domain knowledge) to the system. This transfer of cognitive capability also implies the availability in the intent-based system of capabilities or means for learning, reasoning, and knowledge representation and management. The learning abilities of an intent-based systems can apply to different tasks such as optimization of the intent rendering or intent refinement processes. The fact that an intent-based system is a continuously evolving system creates the condition for continuous learning and optimization. Other cognitive capabilities such as planning can also be leveraged in an intent-based system to anticipate or forecast future system state and response to changes in intents or network conditions and thus elaboration of plans to accommodate the changes while preserving system stability and efficiency in a trade-off with cost and robustness of operations. Cope with unawareness of users (smart recommendations).

5. Capability exposure. Capability exposure consists in the need for expressive network capabilities, requirements, and constraints to be able to compose/decompose intents and map the user's expectations to the system capabilities.

6. Abstract and outcome-driven. Users do not need to be concerned with how intent is achieved and are empowered to think in terms of outcomes. In addition, they do can refer to concepts at a

higher level of abstractions, independent e.g. of vendor-specific renderings.

The described principles are perhaps the most prominent, but they are not an exhaustive list. There are additional aspects to consider, such as:

- Intent targets are not individual devices but typically aggregations (such as groups of devices adhering to a common criteria, such as devices of a particular role) or abstractions (such as service types, service instances, topologies)

- Abstraction and inherent virtualization: agnostic to implementation details

- Human-tailored network interaction: IBN SHOULD speak the language of the user as opposed to requiring the user to speak the language of the device/network

- Explainability as an important IBN function, detection and IBN-aided resolution of conflicting intent, reconciliation of what the user wants and what the network can actually do

- Inherent support, verification, and assurance of trust

All of these principles and considerations have implications on the design of intent-based systems and their supporting architecture and need to be considered when deriving functional and operational requirements.

6. Intent-Based Networking - Functionality

Intent-Based Networking involves a wide variety of functions which can be roughly divided into two categories:

- Intent Fulfillment provides functions and interfaces that allow users to communicate intent to the network, and that perform the necessary actions to ensure that intent is achieved. This includes algorithms to determine proper courses of action and functions that learn to optimize outcomes over time. In addition, it also includes more traditional functions such as any required orchestration of coordinated configuration operations across the network and rendering of higher-level abstractions into lower-level parameters and control knobs.

- Intent Assurance provides functions and interfaces that allow users to validate and monitor that the network is indeed adhering to and complying with intent. This is necessary to assess the

effectiveness of actions taken as part of fulfillment, providing important feedback that allows those functions to be trained or tuned over time to optimize outcomes. In addition, Intent Assurance is necessary to address "intent drift". Intent drift occurs when a system originally meets the intent, but over time gradually allows its behavior to change or be affected until it no longer does, or does so in a less effective manner.

The following sections provide a more comprehensive overview of those functions.

6.1. Intent Fulfillment

Intent fulfillment is concerned with the functions that take intent from its origination by a user (generally, an administrator of the responsible organization) to its realization in the network.

6.1.1. Intent Ingestion and Interaction with Users

The first set of functions is concerned with "ingesting" intent, i.e. obtaining intent through interactions with users. They provide functions that recognize intent from interaction with the user as well as functions that allow users to refine their intent and articulate it in such ways so that it becomes actionable by an Intent-Based System. Typically, those functions go beyond a traditional API, although they may include APIs provided for interactions with other machines. They may support unconventional human-machine interactions, in which a human will not simply give simple commands, but which may involve a human-machine dialog to provide clarifications, to explain ramifications and trade-offs, and to facilitate refinements. The goal is ultimately to make intent-based systems as easy and natural to use as possible, allowing the user to interact with the Intent-Based System in ways that does not involve a steep learning curve forcing the user to learn the "language" of the system

6.1.2. Intent Translation

A second set of functions needs to translate user intent into courses of actions and requests to take against the network, which will be meaningful to network configuration and provisioning systems. These functions lie at the core of Intent-Based Systems, bridging the gap between interaction with users on one hand and the traditional management and operations side that will need to orchestrate provisioning and configuration across the network.

Beyond merely breaking down a higher layer of abstraction (intent) into a lower layer of abstraction (policies, device configuration),

Intent Translation functions can be complemented with functions and algorithms that perform optimizations and that are able to learn and improve over time in order to result in the best outcomes, specifically in cases where multiple ways of achieving those outcomes are conceivable. For example, satisfying an intent may involve computation of paths and other parameters that need will need to be configured across the network. Heuristics and algorithms to do so may evolve over time to optimize outcomes which may depend a myriad of dynamic network conditions and context.

6.1.3. Intent Orchestration

A third set of functions deals with the actual configuration and provisioning steps that need to be orchestrated across the network and that were determined by the previous intent translation step.

6.2. Intent Assurance

Assurance is concerned with the functions that are necessary to ensure that the network indeed complies with the desired intent once it has been fulfilled.

6.2.1. Monitoring

A first set of assurance functions monitors and observes the network and its exhibited behavior. This includes all the usual assurance functions such as monitoring the network for events and performance outliers, performing measurements to assess service levels that are being delivered, generating and collecting telemetry data. Monitoring and observation are required as basis for the next set of functions that assess whether the observed behavior is in fact in compliance with the behavior that is expected based on the intent.

6.2.2. Intent Compliance Assessment

At the core of Intent Assurance are functions that compare the actual network behavior that is being monitored and observed with the intended behavior that is expected per the intent. These functions continuously assess and validate whether the observation indicates compliance with intent. This includes assessing the effectiveness of intent fulfillment actions, including verifying that the actions had the desired effect and assessing the magnitude of the effect as applicable. It can also include functions that perform analysis and aggregation of raw observation data. The results of the assessment can be fed back to facilitate learning functions that optimize outcomes.

Intent compliance assessment also includes assessing whether intent drift occurs over time. Intent drift can be caused by control plane or lower-level management operations that inadvertently cause behavior changes which conflict with intent which was orchestrated earlier. Intent-Based Systems and Networks need to be able to detect when such drift occurs or is about to occur.

6.2.3. Intent Compliance Actions

When intent drift occurs or network behavior is inconsistent with desired intent, functions that are able to trigger corrective actions are needed. This includes actions needed to resolve intent drift and bring the network back into compliance. Alternatively and where necessary, reporting functions need to be triggered that alert operators and provide them with the necessary information and tools to react appropriately, e.g. by helping them articulate modifications to the original intent to moderate between conflicting concerns.

6.2.4. Abstraction, Aggregation, Reporting

The outcome of Intent Assurance needs to be reported back to the user in ways that allows the user to relate the outcomes to their intent. This requires a set of functions that are able to analyze, aggregate, and abstract the results of the observations accordingly. In many cases, lower-level concepts such as detailed performance statistics and observations related to low-level settings need to be "up-leveled" to concepts the user can relate to and take action on.

The required aggregation and analysis functionality needs to be complemented with functions that report intent compliance status and provide adequate summarization and visualization to the user.

7. Life-cycle

Intent is subject to a life-cycle: it comes into being, may undergo changes over the course of time, and may at some point be retracted. This life-cycle is closely tied to various interconnection functions that are associated with the intent concept.

Figure 1 depicts an intent life-cycle and its main functions. The functions were introduced in Section 6 and are divided into two functional (horizontal) planes, reflecting the distinction between fulfillment and assurance. In addition, they are divided into three (vertical) spaces.

The spaces indicate the different perspectives and interactions with different roles that are involved in addressing the functions:

- The user space involves the functions that interface the network and intent-based system with the human user. It involves the functions that allow users to articulate and the intent-based system to recognize that intent. It also involves the functions that report back the status of the network relative to the intent and that allow users to assess whether their intent has the desired effect.

- The translation or Intent-Based System (IBS) space involves the functions that bridge the gap between intent users and network operations. This includes the functions used to translate an intent into a course of action, the algorithms used to plan and optimize those courses of actions also in consideration of feedback, the functions to analyze and abstract observations to validate compliance with the intent and take corrective actions as necessary.

- The Network Operations space, finally, involves the traditional orchestration, configuration, monitoring, and measurement functions, which are used to effectuate the rendered intent and observe its effects on the network.

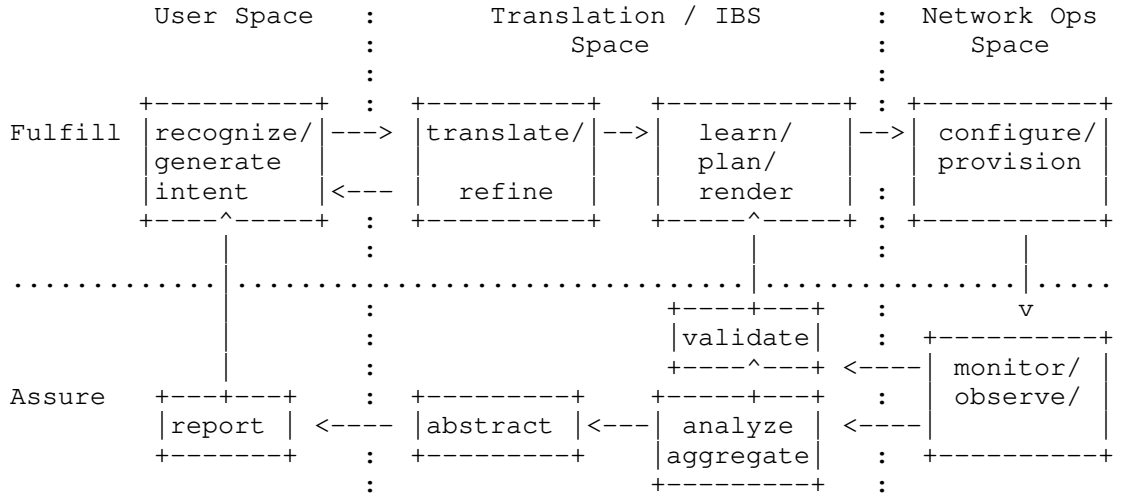


Figure 1: Intent Life-cycle

When carefully inspecting the diagram, it becomes apparent that the intent life-cycle, in fact, involves two cycles, or loops:

- The "inner" intent control loop between IBS and Network Operations space is completely autonomic and does not involve any human in the loop. It involves automatic analysis and validation of intent based on observations from the network operations space. Those observations are fed into the function that plans the rendering of networking intent in order to make adjustments as needed in the configuration of the network.

- The "outer" intent control loop involves the user space and includes the user taking action and adjusting their intent based on feedback from the IBS.

8. Additional Considerations

Given the popularity of the term "intent", its use could be broadened to encompass also known or related concepts, resulting in "intent-washing" that paints those concepts in a new light by simply applying new intent terminology to them. However, in some cases, this actually makes sense not just a marketing ploy but as a way to better relate previously existing and new concepts.

In that sense and regards to intent, it make sense to distinguish various subcategories of intent as follows:

- Operational Intent: defines intent related to operational goals of an operator; corresponds to the original "intent" term and the concepts defined in this document.

- Rule Intent: a synonym for policy rules regarding what to do when certain events occur.

- Service intent: a synonym for customer service model [RFC8309].

- Flow Intent: A synonym for a Service Level Objective for a given flow.

A comprehensive set of classifications of different concepts and categories of intent will be described in a separate document.

9. IANA Considerations

Not applicable

10. Security Considerations

This document describes concepts and definitions of Intent-based Networking. As such, the below security considerations remain high level, i.e. in the form of principles, guidelines or requirements.

More detailed security considerations will be described in the documents that specify the architecture and functionality.

Security in Intent-based Networking can apply to different facets:

- Securing the intent-based system itself.
- Mitigating the effects of erroneous, harmful or compromised intents.
- Expressing security policies or security-related parameters with intents.

Securing the intent-based system aims at making the intent-based system operationally secure by implementing security mechanisms and applying security best practices. In the context of Intent-based Networking, such mechanisms and practices may consist in intent verification and validation; operations on intents by authenticated and authorized users only; protection against or detection of tampered intents. Such mechanisms may also include the introduction of multiple levels of intent. For example, intent related to securing the network should occur at a "deeper" level that overrides other levels of intent if necessary, and that is not subject to modification through regular operations but through ones that are specifically secured. Use of additional mechanisms such as explanation components that describe the security ramifications and trade-off should be considered as well.

Mitigating the effects of erroneous or compromised intents aims at making the intent-based system operationally safe by providing checkpoint and safeguard mechanisms and operating principles. In the context of Intent-based Networking, such mechanisms and principles may consist in the ability to automatically detect unintended, detrimental or abnormal behavior; the ability to automatically (and gracefully) rollback or fallback to a previous "safe" state; the ability to prevent or contain error amplification (due to the combination of higher degree of automation and the intrinsic higher degree of freedom, ambiguity, and implicit conveyed by intents); dynamic levels of supervision and reporting to make the user aware of the right information, at the right time with the right level of context. Erroneous or harmful intents may inadvertently propagate and compromise security. In addition, compromised intents, for example intent forged by an inside attacker, may sabotage or harm the network resources and make them vulnerable to further, larger attacks, e.g. by defeating certain security mechanisms.

Expressing security policies or security-related parameters with intents consists in using the intent formalism (a high-level,

declarative abstraction), or part(s) of an intent statement to define security-related aspects such as data governance, level(s) of confidentiality in data exchange, level(s) of availability of system resources, of protection in forwarding paths, of isolation in processing functions, level(s) of encryption, authorized entities for given operations, etc.

The development and introduction of Intent-Based Networking in operational environments will certainly create new security concerns. Such security concerns have to be anticipated at the design and specification time. However, Intent-Based Networking may also be used as an enabler for better security. For instance, security and privacy rules could be expressed in more human-friendly and generic way and be less technology-specific and less complex, leading to fewer low-level configuration mistakes. The detection of threats or attacks could also be made more simple and comprehensive thanks to conflict detection at higher-level or at coarser granularity

More thorough security analyses should be conducted as our understanding of Intent-Based Networking technology matures.

11. References

11.1. 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>.

11.2. Informative References

[Boutaba07] Boutaba, R. and I. Aib, "Policy-Based Management: A Historical perspective. Journal of Network and Systems Management (JNSM), Springer, Vol. 15 (4).", December 2007.

[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-04 (work in progress), July 2020.

[Lenrow15]
Lenrow, D., "Intent As The Common Interface to Network Resources, Intent Based Network Summit 2015 ONF Boulder: IntentNBI", February 2015.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <https://www.rfc-editor.org/info/rfc7575>.

[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>.

[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>.

[Sloman94]
Sloman, M., "Policy Driven Management for Distributed Systems. Journal of Network and Systems Management (JNSM), Springer, Vol. 2 (4).", December 1994.

[Strassner03]
Strassner, J., "Policy-Based Network Management. Elsevier.", 2003.

[TR523] Foundation, O. N., "Intent NBI - Definition and Principles. ONF TR-523.", October 2016.

Authors' Addresses

Alexander Clemm
Futurewei
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ludwig@clemm.org

Laurent Ciavaglia
Nokia
Route de Villejust
Nozay 91460
FR

Email: laurent.ciavaglia@nokia.com

Lisandro Zambenedetti Granville
Federal University of Rio Grande do Sul (UFRGS)
Av. Bento Goncalves
Porto Alegre 9500
BR

Email: granville@inf.ufrgs.br

Jeff Tantsura
Apstra, Inc.

Email: jefftant.ietf@gmail.com

Network Working Group
Internet Draft
Intended status: Informational
Expires: May 2021

C. Li
China Telecom
O. Havel
W. Liu
A. Olariu
Huawei Technologies
P. Martinez-Julia
NICT
J. Nobre
UFRGS
D. Lopez
Telefonica, I+D

November 2, 2020

Intent Classification
draft-irtf-nmrg-ibn-intent-classification-01

Abstract

RFC7575 defines Intent as an abstract high-level policy used to operate the network. Intent management system includes an interface for users to input requests and an engine to translate the intents into the network configuration and manage their life-cycle. Up to now, there is no commonly agreed definition, interface or model of intent.

This document discusses mostly the concept of network intents, but other types of intents are also being considered. Specifically, it highlights stakeholder perspectives of intent, methods to classify and encode intent, the associated intent taxonomy, and defines relevant intent terms where necessary. This document provides a foundation for intent related research and facilitate solution development.

Table of Contents

1. Introduction

The vision of intent-driven networks has attracted a lot of attention, as it promises to simplify the management of networks by human operators. This is done by simply specifying what should happen on the network, without giving any instructions on how to do it. This

promise led many telecom companies to begin adopting this new vision, and many Standards Development Organization (SDOs) to propose different intent framework.

Several SDOs and open source projects, such as Internet Engineering Task Force (IETF) (by the Autonomic Networking Integrated Model and Approach Working Group [ANIMA]), Open Networking Foundation (ONF) [ONF], Open Network Operating System (ONOS) [ONOS], have proposed intents for defining a set of network operations to execute in a declarative manner.

IETF [ANIMA] defines intent as a declarative policy, but still lacks a more complete definition, a tentative format, and a life-cycle. Within ONOS [ONOS], intent is represented as a list of Command-Line Interface (CLI) commands that allows users to bypass low-level details on the network, such as flows or host addresses. ONF through its Boulder and Aspen projects focuses on Northbound Interface (NBI) semantics and intent models.

The SDOs usually came up with their own way of specifying an intent, and with their own understanding of what an intent is. Besides that, each SDO defines a set of terms and level of abstraction, its intended users, and the applications and usage scenarios.

However, most intent approaches proposed by SDOs share the same following features:

- It must be declarative in nature, meaning that a user specifies the goal on the network without specifying how to achieve that goal.
- It must be vendor agnostic, in the sense that it abstracts the network capabilities, or the network infrastructure from the user, and it can be ported across different platforms.
- It must provide an easy-to-use interface, which simplifies the users' interaction with the intent system through the usage of familiar terminology or concepts.
- It should be able to detect and resolve intent conflicts, which include, for example, static (compile-time) conflicts and dynamic (run-time) conflicts.

Currently, work is underway on unifying a common understanding of intent concepts and terminology. Concerning NMRG, [CLEMM] is a document to present a definition for intent as higher-level declarative policy that operates at the level of network and services

it provides. In addition, this document captures the differences between intent, policy and service.

However, even with proposed intent concepts and terminology, as well as agreement on common intent characteristics, an intent may still be viewed in different ways by different stakeholders for different use cases and solutions. This document mostly addresses intents in the context of network intents, however other types of intents are not excluded, as presented in Section 4.4. and Section 6.2. .

The goal of this document is to clarify what an intent represents for different stakeholders through a classification on various dimensions, such as solutions, users, and intent types. This classification can ensure a common understanding across all participants and be used to identify the scope and priorities of individual projects, Proof of Concepts (PoCs), research initiatives, or open-source projects. This goal is achieved by proposing the methodology and initial classification tables. This methodology can be used to update the tables by adding or removing different solutions, users or intent types in order to cater for future scenarios, applications or domains.

The present document, together with [CLEMM], aims to become the foundation for future intent-related topic discussions regarding the NMRG.

2. Key Words

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Acronyms

AI: Artificial Intelligence

API: Application Programming Interface

CE: Customer Equipment

CFS: Customer Facing Service

CLI: Command Line Interface

DB: Data Base

DC: Data Center

ECA: Event-Condition-Action

GBP: Group-Based Policy

IETF: Internet Engineering Task Force

IP: Internet Protocol

O&M: Operations & Maintenance

ONF: Open Networking Foundation

ONOS: Open Network Operating System

PNF: Physical Network Function

QoS: Quality of Service

RFS: Resource Facing Service

SDO: Standards Development Organization

SD-WAN: Software-Defined Wide-Area Network

SLA: Service-Level Agreement

SUPA: Simplified Use of Policy Abstractions

VLAN: Virtual Local Area Network

VM: Virtual Machine

VPN: Virtual Private Network

4. Abstract Intent Requirements

In order to understand the different intent requirements that would drive intent classification, we first need to understand what intent means for different intent users.

4.1. What is Intent?

The term Intent has become very widely used in the industry for different purposes, sometimes it is not even in agreement with SDO shared principles mentioned in the Introduction.

Different stakeholders consider an intent to be an ECA policy, a GBP policy, a business policy, a network service, a customer service, a network configuration, application/application group policy, any operator/administrator task, network troubleshooting/diagnostics/ test, a new app, a marketing term for existing management/orchestration capabilities, etc. Their intent is sometimes technical, non-technical, abstract or technology specific. For some stakeholders, intent is a subset of these and for other stakeholders intent is all of these. It has in some cases become a term to replace a very generic 'service' or 'policy' terminology.

Concerning this, [CLEMM] draft brings clarification with relation to what an intent is and how it differentiates from policies and services. Future versions of this draft will be kept aligned with [CLEMM].

While it is easier for those familiar with different standards to understand what service, CFS, RFS, resource, policy continuum, ECA policy, declarative policy, abstract policy or intent policy is, it may be more difficult for the wider audience.

An intent is mistaken by many to be just a synonym for policy. While it is easier for those familiar with different standards to understand what service, CFS, RFS, resource, policy continuum, ECA policy, declarative policy, abstract policy or intent policy is, it may be more difficult for the wider audience. Furthermore, those familiar with policies understand the difference between a business, intent, declarative, imperative, and ECA policy.

Therefore, it is important to start a discussion in the industry about what intent is for different solutions and intent users. It is also imperative to try to propose some intent categories/

classifications that could be understood by a wider audience. This would help us define intent interfaces, DSLs and models.

4.2. Intent Solutions and Intent Users

Different Solutions and Actors have different requirements, expectations and priorities for intent-driven networking. They require different intent types and have different use cases. Some users are more technical and require intents that expose more technical information. Other users do not understand networks and require intents that shield them from different networking concepts and technologies. The following are the solutions and intent users that intent-driven networking needs to support:

Solutions	Intent Users
Carrier Networks	Network Operator Service Designers Service Operators Customers/Subscribers
DC Networks	Cloud Administrator Underlay Network Administrator App Developers End-Users
Enterprise Networks	Enterprise Administrator App Developers End-Users

These intent solutions and intent users represent a starting point for the classification and are expendable through the methodology presented in Section 6.1. .

- For carrier networks scenario, for example, if the end-users wants to watch high-definition video, then the intent is to convert the video image to 1080p rate for the users.

- For DC networks scenario, administrators have their own clear network intent such as load balancing. For all traffic flows that need NFV service chaining, restrict the maximum load of any VNF node/container below 50% and the maximum load of any network link below 70%.

- For Enterprise Networks scenario, enterprise administrators express their intent from an external client (application service provider). For example, when hosting a video conference, multiple remote accesses are required. An example of the intent expressed to the network operator is: For any user of this application, the arrival time of hologram objects of all the remote tele-presenters should be synchronised within 50ms to reach the destination viewer for each conversation session.

4.3. Current Problems and Requirements

Network APIs and CLIs are too complex due to the fact that they expose technologies and topologies. App developers and end-users do not want to set IP Addresses, VLANs, subnets, ports, etc. Operators and administrators would also benefit from the simpler interfaces, like:

- Allow Customer Site A to be connected to Internet via Network B
- Allow User A to access all internal resources, except the Server B
- Allow User B to access Internet via Corporate Network A
- Move all Users from Corporate Network A to the Corporate Network B
- Request Gold VPN service between my sites A, B and C
- Provide CE Redundancy for all Customer Sites
- Add Access Rules to my Service

Networks are complex, with many different protocols and encapsulations. Some basic questions are not easy to answer:

- Can User A talk to User B?
- Can Host A talk to Host B?
- Are there any loops in my network?
- Are Network A and Network B connected?
- Can User A listen to communications between Users B and C?

Operators and Administrators manually troubleshoot and fix their networks and services. They instead want:

- a reliable network that is self-configured and self-assured based on the intent
- to be notified about the problem before the user is aware
- automation of network/service recovery based on intent (self-healing, self-optimization)
- to get suggestions about correction/optimization steps based on experience (historical data and behaviour)

Therefore, Operators and Administrators want to:

- simplify and automate network operations
- simplify definitions of network services
- provide simple customer APIs for Value Added Services (operators)
- be informed if the network or service is not behaving as requested
- enable automatic optimization and correction for selected scenarios
- have systems that learn from historic information and behaviour

End-users cannot build their own services and policies without becoming technical experts and they must perform manual maintenance actions. Application developers and end-users/subscribers want to be able to:

- build their own network services with their own policies via simple interfaces, without becoming networking experts
- have their network services up and running based on intent and automation only, without any manual actions or maintenance

4.4. Intent Types that need to be supported

The following intent types need to be supported, in order to address the requirements from different solutions and intent users:

- Customer service intent
    - for customer self-service with SLA or add a service
    - for service operator orders

- Network and Underlay Network service intent
  - for service operator orders
  - for intent driven network configuration, verification, correction and optimization
  - for intent created and provided by the underlay network administrator
- Network and Underlay Network intent
  - For network configuration
  - For automated lifecycle management of network configurations
  - For network resources (switches, routers, routing, policies, underlay)
- Cloud management intent
  - For DC configuration, VMs, DB Servers, APP Servers
  - For communication between VMs
- Cloud resource management intent
  - For cloud resource life-cycle management (policy driven self-configuration and auto-scaling and recovery/optimization)
- Strategy intent
  - For security, QoS, application policies, traffic steering, etc.
  - For configuring and monitoring policies, alarms generation for non-compliance, auto-recovery
  - For design models and policies for network and network service design
  - For design workflows, models and policies for operational task intents
- Operational task intents
  - For network migration

- For server replacements
- For device replacements
- For network software upgrades
- To automate any tasks that operators/administrator often perform

o Intents that affect other intents

- It may be task-based intent that modifies many other intents.
- The task itself is short-lived, but the modification of other intents has an impact on their life-cycle, so those changes must continue to be continuously monitored and self-corrected/self-optimized.

5. Functional Characteristics and Behaviour

Intent can be used to operate immediately on a target (much like issuing a command), or whenever it is appropriate (e.g., in response to an event). In either case, intent has a number of behaviours that serve to further organize its purpose, as described by the following subsections.

5.1. Abstracting Intent Operation

The modelling of Intents can be abstracted using the following three-tuple:

{Context, Capabilities, Constraints}

- Context grounds the intent, and determines if it is relevant or not for the current situation. Thus, context selects intents based on applicability.
- Capabilities describe the functionality that the intent can perform. Capabilities take different forms, depending on the expressivity of the intent as well as the programming paradigm(s) used.
- Constraints define any restrictions on the capabilities to be used for that particular context.

Metadata can be attached via strategy templates to each of the elements of the three-tuple, and may be used to describe how the

intent should be used and how it operates, as well as prescribe any operational dependencies that must be taken into account.

5.2. Intent User Types

Intent user types, or intent actors as they are known in the area of declarative policy, represent the users that define and issue the intent request. Depending on the Intent Solutions, there are specific intent actors. Examples of intent actors are customers, network operators, service operators, enterprise administrators, cloud administrators, and underlay network administrators, or application developers.

- Customers and end-users do not necessarily know the functional and operational details of the network that they are using. Furthermore, they lack skills to understand such details; in fact, such knowledge is typically not relevant to their job. In addition, the network may not expose these details to its users. This class of actor focuses on the applications that they run, and uses services offered by the network. Hence, they want to specify policies that provide consistent behaviour according to their business needs. They do not have to worry about how the intents are deployed onto the underlying network, and especially, whether the intents need to be translated to different forms to enable network elements to understand them.

- Application developers work in a set of abstractions defined by their application and programming environment(s). For example, many application developers think in terms of objects (e.g., a VPN). While this makes sense to the application developer, most network devices do not have a VPN object per se; rather, the VPN is formed through a set of configuration statements for that device in concert with configuration statements for the other devices that together make up the VPN. Hence, the view of application developers matches the services provided by the network, but may not directly correspond to other views of other actors.

- Management personnel, such as network operators, may have the knowledge of the underlying network. However, they may not understand the details of the applications and services of Customers and End-Users.

5.3. Intent Scope

Intent are used to manage the behaviour of the networks they are applied to and all intents are applied within a specific scope, such as:

- o Connectivity scope, if the intent creates or modifies a connection.
- o Security scope, if the intent specifies the security characteristics of the network or users.
- o Application scope, when the intent specifies the applications to be affected by the intent request.
- o QoS Scope, when the intent specifies the QoS characteristics of the network.

These intent scopes are expendable through the methodology presented in Section 6.1. .

5.4. Intent Network Scope

Regardless on the intent user type, their intent request is affecting the network, or network components, which are representing the intent targets.

Thus, intent network scope, or policy target as known in the area of declarative policy, can represent VNFs or PNFs, Physical Network Elements, Campus networks, SD-WAN networks, radio access networks, cloud edge, cloud core, branch, etc.

5.5. Intent Abstraction

Intent can be classified by whether it is necessary to feedback technical network information or non-technical information to the intended proponent after the intent is executed. As well, intent abstraction covers the level of technical details in the intent itself.

- o For ordinary users, they do not care how the intent is executed, or the details of the network. As a result, they do not need to know the configuration information of the underlying network. They only focus on whether the intent execution result achieves the goal, and the execution effect such as the quality of completion and the length of execution. In this scenario, we refer to an abstraction without technical feedback.

- For administrators, such as network administrators, they perform intents, such as allocating network resources, selecting transmission paths, handling network failures, etc. They require multiple feedback indicators for network resource conditions, congestion conditions, fault conditions, etc. after execution. In this case, we refer to an abstraction with technical feedback.

As per intent definition provided in [CLEMM], lower-level intents are not considered to qualify as intents. However, we kept this classification to identify any PoCs/Demos/Use Cases that still either require or implement lower level of abstraction for intents.

5.6. Intent Life-cycle

Intents can be classified into transient and persistent intents:

- If intent is transient, it has no life-cycle management. As soon as the specified operation is successfully carried out, the intent is finished, and can no longer affect the target object.

- If the intent is persistent, it has life-cycle management. Once the intent is successfully activated and deployed, the system will keep all relevant intents active until they are deactivated or removed.

5.7. Hierarchy

In different phases of the autonomous driving network [TMF-auto], the intents are different. A typical example of autonomous driving network Level 0 to 5 are listed as below.

- Level 0 - Traditional manual network: O&M personnel manually control the network and obtain network alarms and logs. - No intent

- Level 1 - Partially automated network: Automated scripts are used to automate service provisioning, network deployment, and maintenance. Shallow perception of network status and decision making suggestions of machine; - No intent

- Level 2 - Automated network: Automation of most service provisioning, network deployment, and maintenance comprehensive perception of network status and local machine decision making; - simple intent on service provisioning

- Level 3 – Self-optimization network: Deep awareness of network status and automatic network control, meeting users' network intentions. – Intent based on network status cognition

- Level 4 – Partial autonomous network: In a limited environment, people do not need to participate in decision-making and adjust themselves. – Intent based on limited AI

- Level 5 – Autonomous network: In different network environments and network conditions, the network can automatically adapt to and adjust to meet people's intentions. – Intent based on AI

6. Intent Classification

This chapter proposes an intent classification approach that may help to classify mainstream intent related demos/tools.

The three classifications in this draft have been proposed from scratch, following the methodology presented, through three iterations: one for carrier Intent Solution, one for DC Intent Solution, and one for enterprise Intent Solution. For each Intent solution, we identified the specific Intent Users and Intent Types. Then, we further identified the Intent Scope, Network Scope, Abstractions, and Life-cycle requirements.

These classifications and the generated tables can be easily extended. For example, for the DC Intent Solution, a new category is identified, i.e. Resource Scope, and the classification table has been extended accordingly.

In the future, as new scenarios, applications, and domains are emerging, new classifications and taxonomies can be identified, following the proposed methodology.

The output of the intent classification is the intent taxonomy introduced in the next sections.

Thus, this section first introduces the proposed intent classification methodology, followed by consolidated intent taxonomy for three intent solutions, and then by concrete examples of intent classifications for three different intent solutions (e.g. Carrier Network, Data Center, and Enterprise) that were derived using the proposed methodology and then can be filled in for PoCs, demos, research projects or future drafts.

6.1. Intent Classification Methodology

This section describes the methodology used to derive the initial classification proposed in the draft. The proposed methodology can be used to create new intent classifications from scratch, by analysing the solution knowledge. As well, the methodology can be used to update existing classification tables by adding or removing different solutions, users or intent types in order to cater for future scenarios, applications or domains.

The intent classification workflow starts from the Solution Knowledge, which can provide information on requirements, use cases, technologies used, network properties, actors that define and issue the intent request, and requirements. The following, defines the steps to classify an intent:

1. The information provided in the solution knowledge is provided as input to identifying the intent solution (e.g. Carrier, Enterprise, and Data Center). This intent solution is reviewed against the existing classification and it can either be used or add/remove the intent solution identified from the solution knowledge (R1-U1).

2. The next step is identifying the intent user types (e.g. customer, network operators, service operators, etc.) and then review existing classification and use it or add/remove the intent user type identified from the solution knowledge (R2-U2).

3. The next phase is to identify the type of intent (e.g. Network Intent, Customer Service Intent) and then review existing classification and use/add/remove the intent type (R3-U3).

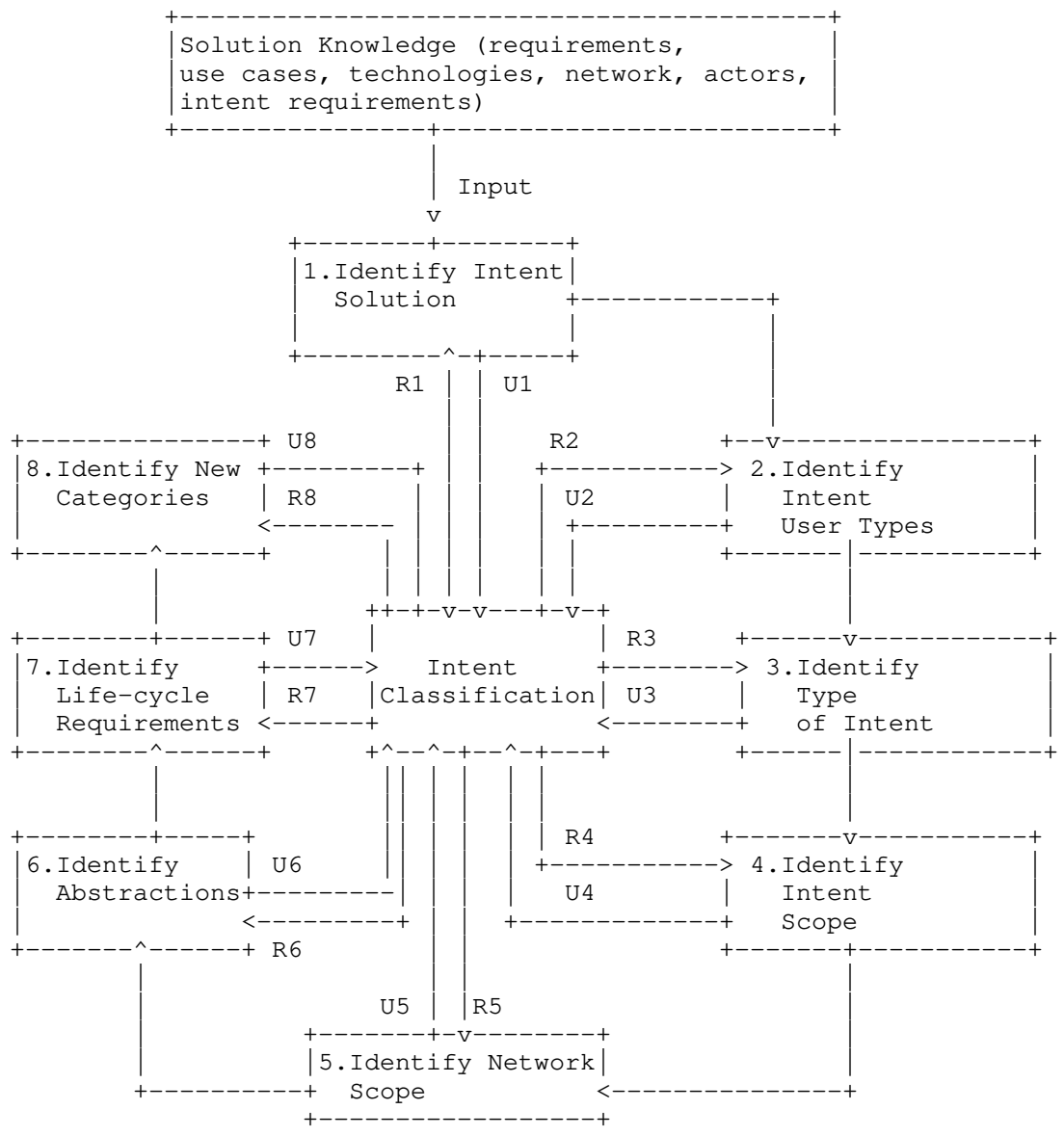
4. The forth step is identifying the intent scope (e.g. Connectivity, Application) based on the Solution Knowledge and then review existing classification and use/add/remove the identified intent scope (R4-U4).

5. The next step is to identify the network scope (e.g. Campus, Radio Access) and then then review existing classification and either use it or add/remove the identified network scope (R5-U5).

6. The next phase is to identify the abstractions (e.g. technical, non-technical) and then review existing classification and use/add/remove the abstractions (R6-U6).

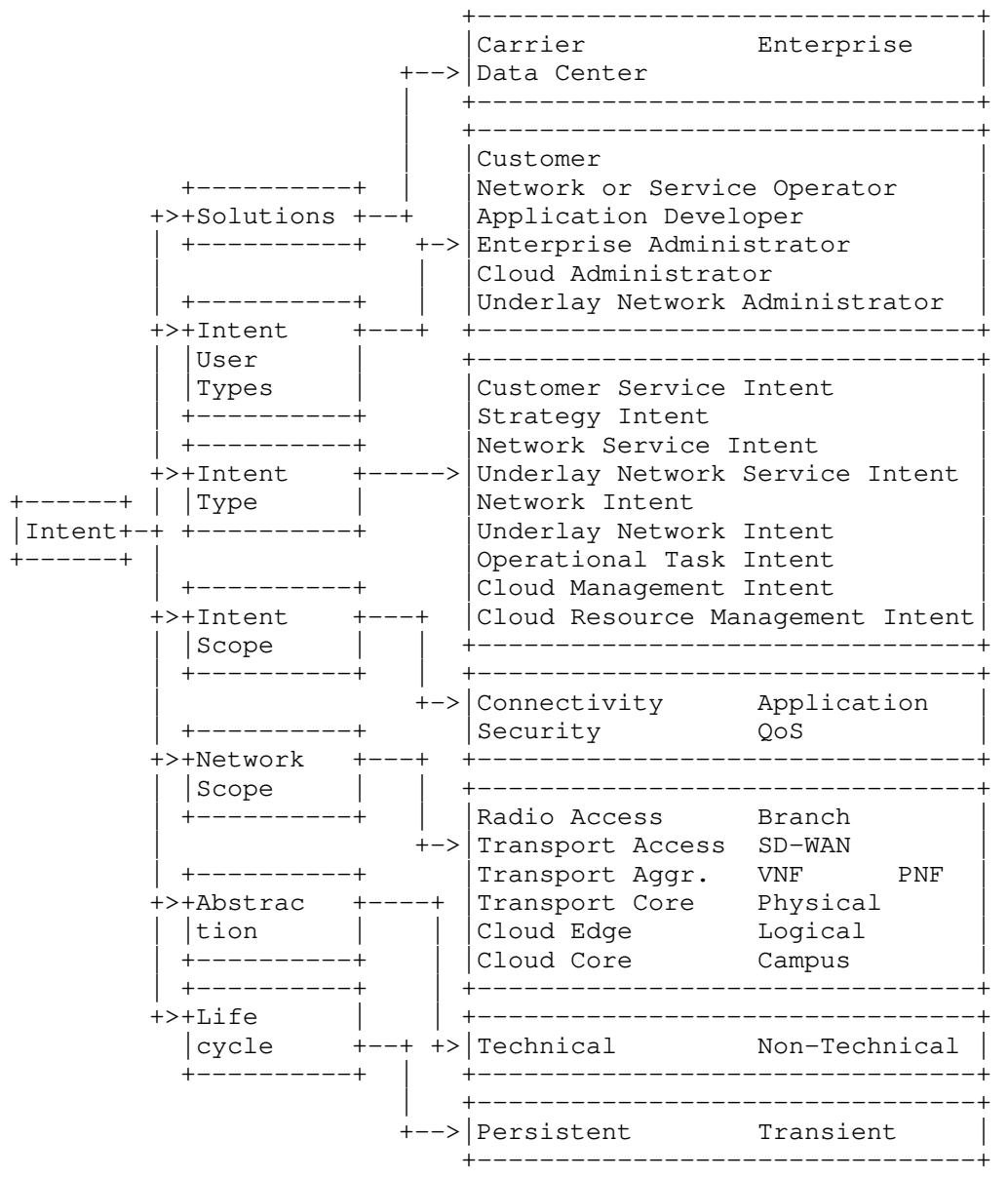
7. The seventh step is to identify the life-cycle requirements (e.g. persistent, transient) and then review existing classification and use/add/remove the life-cycle requirements (R7-U7).

8. The last step is to identify any new categories and use/add the newly identified categories. New categories can be identified as new domains or applications are emerging, or new areas of concern (e.g. privacy, compliance) might arise, which are not listed in the current methodology.


Solution Knowledge (requirements,
use cases, technologies, network, actors,
intent requirements)
Input
1.Identify Intent
Solution
R1
U1
U8
R2
8.Identify New
Categories
R8
U2
2.Identify
Intent
User Types
U7
R3
7.Identify
Life-cycle
Requirements
R7
Intent
Classification
U3
3.Identify
Type
of Intent
6.Identify
Abstractions
U6
R4
U4
4.Identify
Intent
Scope
R6
U5
R5
5.Identify Network
Scope

6.2. Intent Taxonomy

The following taxonomy describes the various intent solutions, intent user types, intent types, intent scopes, network scopes, abstractions and life-cycle and represents the output of the intent classification tables for each of the solutions addressed (i.e. Carrier Solution, Data Center, and Enterprise).


Carrier Enterprise
Data Center
Customer
Network or Service Operator
Application Developer
Enterprise Administrator
Cloud Administrator
Underlay Network Administrator
Solutions
Intent User Types
Customer Service Intent
Strategy Intent
Network Service Intent
Intent Type
Underlay Network Service Intent
Network Intent
Underlay Network Intent
Operational Task Intent
Cloud Management Intent
Intent
Cloud Resource Management Intent
Intent
Scope
Connectivity
Application
Security
QoS
Network
Scope
Radio Access
Branch
Transport Access
SD-WAN
Transport Aggr.
VNF
PNF
Abstrac
tion
Transport Core
Physical
Cloud Edge
Logical
Cloud Core
Campus
Life
cycle
Technical
Non-Technical
Persistent
Transient
Intent

6.3. Intent Classification for Carrier Solution

Users and Intent Types

The following table describes the Intent Users in Carrier Solutions and Intent Types with their descriptions for different intent users.

Intent User	Intent Type	Intent Type Description
Customer/ Subscriber	Customer Service Intent	Customer Self-Service with SLA and Value Added Service Example: Always maintain high quality of service and high bandwidth for gold level users. Operational statement: Measure the network congestion status, give different adaptive parameters to stations of different priority, thus in heavy load situation, makes the bandwidth of the high-priority users guaranteed. At the same time ensure the overall utilization of system, improve the overall throughput of the system.
	Strategy Intent	Customer designs models and policy intents to be used by Customer Service Intents. Example: Request reliable service during peak traffic periods for apps of type video.
Network Operator	Network Service Intent	Service provided by Network Service Operator to the Customer (e.g. the Service Operator) Example: Request network service with delay guarantee for access customer A.
	Network Intent	Network Operator requests network-wide (service underlay or other network-wide configuration) or network resource configurations (switches, routers,

			routing, policies). Includes Connectivity, Routing, QoS, Security, Application Policies, Traffic Steering Policies, Configuration policies, Monitoring policies, alarm generation for non-compliance, auto-recovery, etc. Example: Request high priority queueing for traffic of class A.
		Operational Task Intent	Network Operator requests execution of any automated task other than Network Service Intent and Network Intent (e.g. Network Migration, Server Replacements, Device Replacements, Network Software Upgrades). Example: Request migration of all services in Network N to backup path P.
		Strategy Intent	Network Operator designs models, policy intents and workflows to be used by Network Service Intents, Network Intents and Operational Task Intents. Workflows can automate any tasks that Network Operator often performed in addition to Network Service Intents and Network Intents Example: Ensure the load on any link in the network is not higher than 50%.

Service Operator	Customer Service Intent	Service Operator's Customer Orders, Customer Service / SLA Example: Provide service S with guaranteed bandwidth for customer A.
	Network Service Intent	Service Operator's Network Orders / Network SLA Example: Provide network guarantees in terms of security, low latency and high bandwidth
	Operational Task Intent	Service Operator requests execution of any automated task other than Customer Service Intent and Network Service Intent Example: Update service operator portal platforms and their software regularly. Move services from Network Operator 1 to Network Operator 2.
	Strategy Intent	Service Operator designs models, policy intents and workflows to be used by Customer Service Intents, Network Service Intents and Operational Task Intents. Workflows can automate any tasks that Service Operator often performed in addition to Network Service Intents and Network Intents. Example: Request network service guarantee to avoid network congestion during special periods such as Black Friday, and Christmas.
Application Developer	Customer Service Intent	Customer Service Intent API provided to the Application Developers Example: API to request network to watch HD video 4K/8K.

		Network Service Intent	Network Service Intent API provided to the Application Developers Example: API to request network and monitoring and traffic grooming.
		Network Intent	Network Intent API provided to the Application Developers Example: API to request network resources configuration.
		Operational Task Intent	Operational Task Intent API provided to the Application Developers. This is for the trusted internal Operator / Service Providers / Customer DevOps Example: API to request server migrations.
		Strategy Intent	Application Developer designs models, policy and workflows to be used by Customer Service Intents, Network Service Intents and Operational Task Intents. This is for the trusted internal Operator/Service Provider/ Customer DevOps Example: API to design network load balancing strategies during peak times

Categories

The following are the proposed categories:

Intent Scope: C1=Connectivity, C2=Security, C3=Application, C4=QoS

Network Scope:

- Network Domain: C1=Radio Access, C2=Transport Access, C3=Transport Aggregation, C4=Transport Core, C5=Cloud Edge, C6=Cloud Core)
- Network Function (NF) Scope: C1=VNFs, C2=PNFs

Abstraction(ABS): C1=Technical(with technical feedback), C2=Non-technical (without technical feedback) see Section 5.2. .

Life-cycle (L-C): C1=Persistent (Full life-cycle), C2=Transient (Short Lived)

Classification Example

This section depicts an example on how the methodology described in Section 6.1. can be used in order to classify intents introduced in the 'A Multi-Level Approach to IBN' PoC demonstration [POC-IBN]. The PoC considered two intents: slice intents and service chain intents.

In this PoC [POC-IBN], a slice intent expresses a request for a network slice with two types of components: a set of top layer virtual functions, and a set of virtual switches and/or routers of L2/L3 VNFs. A service chain intent expressed a request for a service operated through a chain of service components running in L4-L7 virtual functions.

Following the intent classification methodology described step-by-step in Section 6.1. , we identify the following:

1. The Intent Solution is for the Carrier.

2. The Intent User Type is the Network Operator for the slice intent, and the Service Operator for the service chain intent

3. The Type of Intent, is a Network Service Intent for the slice intent, and a Customer Service Intent for the service chain intent.

4. The Intent Scopes are connectivity and application.

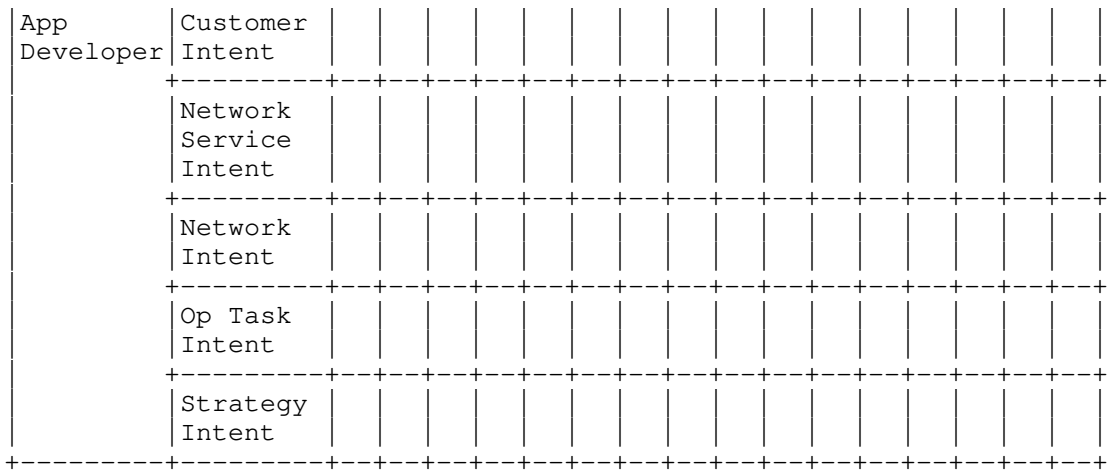
5. The Network Scope is a logical one.

6. The Abstractions are with technical feedback for the slice intent, and without technical feedback for the service chain intent

7. The life-cycle is persistent.

The following table shows how to represent this information in a tabular form. The 'X' in the table refers to the slice intent, and the 'Y' in the table refers to the service chain intent.

Intent User	Intent Type	Intent Scope				NF Scope		Network Scope						ABS		L-C	
		C1	C2	C3	C4	C1	C2	C1	C2	C3	C4	C5	C6	C1	C2	C1	C2
Customer / Sub-scriber	Customer Service Intent																
	Strategy Intent																
Network Operator	Network Service Intent	X		X		X						X	X	X		X	
	Network Intent																
	Operational Task Intent																
	Strategy Intent																
Service Operator	Customer Service Intent	Y		Y		Y							Y	Y	Y	Y	
	Network Service Intent																
	Op Task Intent																
	Strategy Intent																


App Developer
Customer Intent
Network Service Intent
Network Intent
Op Task Intent
Strategy Intent

6.4. Intent Classification for Data Center Solutions

Users and Intent Types

The following table describes the Intent Users in DCN Solutions and Intent Types with their descriptions for different intent users.

Intent User	Intent Type	Intent Type Description
Customer / Tenants	Customer Service Intent	Customer Self-Service via Tenant Portal, Customers may have multiple type of end-users. Example: Request GPU computing and storage resources to meet 10k video surveillance services.
	Strategy Intent	This includes models and policy intents designed by Customers/ Tenants to be used by Customer and End-User Intents. Example: Request dynamic computing and storage resources of the service in special and daily times.
Cloud Administrator	Cloud Management Intent	Configuration of VMs, DB Servers, App Servers, Connectivity, Communication between VMs. Example: Request connectivity between VMs A,B,and C in Network N1.
	Cloud Resource Management Intent	Policy-driven self-configuration and and recovery / optimization Example: Request automatic life -cycle management of VM cloud resources.
	Operational Task Intent	Cloud Administrator requests execution of any automated task other than Cloud Management Intents and Cloud Resource Management Intents. Example: Request upgrade operating system to version X on all VMs in Network N1.

		Operational statement: an intent to update a system might reconfigure the system topology (connect to a service and to peers), exchange data (update the content), and uphold a certain QoE level (allocate sufficient network resources). The network,thus, carries out the necessary configuration to best serve such an intent; e.g. setting up direct connections between terminals, and allocating fair shares of router queues considering other network services.
	Strategy Intent	Cloud Administrator designs models, policy intents and workflows to be used by other intents. Automate any tasks that Administrator often performs, in addition to life-cycle of Cloud Management Intents and Cloud Management Resource Intents. Example: In case of emergency, automatically migrate all cloud resources to DC2.
Underlay Network Administrator	Underlay Network Service Intent	Service created and provided by the Underlay Network Administrator. Example: Request underlay service between DC1 and DC2 with bandwidth B.
	Underlay Network Intent	Underlay Network Administrator requests some DCN-wide underlay network configuration or network resource configurations. Example: Establish and allocate DHCP address pool.
	Operational Task Intent	Underlay Network Administrator requests execution of the any automated task other than Underlay Network Service and Resource

		Intent. Example: Request automatic rapid detection of device failures and pre-alarm correlation.
	Strategy Intent	Underlay Network Administrator designs models, policy intents & workflows to be used by other intents. Automate any tasks that Administrator often performs Example: For all traffic flows that need NFV service chaining, restrict the maximum load of any VNF node/container below 50% and the maximum load of any network link below 70%.
Application Developer	Cloud Management Intent	Cloud Management Intent API provided to the Application Developers. Example: API to request configuration of VMs,or DB Servers
	Cloud Resource Management Intent	Cloud Resource Management Intent API provided to the Application Developers. Example: API to request automatic life-cycle management of cloud resources.
	Underlay Network Service Intent	Underlay Network Service API provided to the Application Developers. Example: API to request real-time monitoring of device condition.
	Underlay Network Intent	Underlay Network Resource API provided to the Application Developers. Example: API to request dynamic management of IPv4 address pool resources.

		Operational Task Intent	Operational Task Intent API provided to the trusted Application Developer (internal DevOps). Example: API to request automatic rapid detection of device failures and pre-alarm correlation
		Strategy Intent	Application Developer designs models, policy intents and building blocks to be used by other intents. This is for the trusted internal DCN DevOps. Example: API to request load balancing thresholds.

Categories

The following are the proposed categories:

Intent Scope: C1=Connectivity, C2=Security, C3=Application, C4=QoS C5=Storage C6=Compute

Network Scope

- o Network Domain: DC Network
- o DCN Network (DCN Net) Scope: C1=Logical, C2=Physical
- o DCN Resource (DCN Res) Scope: C1=Virtual, C2=Physical

Abstraction(ABS): C1=Technical(with technical feedback), C2=Non-technical (without technical feedback), see Section 5.2.

Life-cycle (L-C): C1=Persistent (Full life-cycle), C2=Transient (Short Lived)

Classification Example

This section depicts an example on how the methodology described in Section 6.1. can be used in order to classify intents introduced in the 'A Multi-Level Approach to IBN' PoC demonstration [POC-IBN]. The PoC considered two intents: slice intents and service chain intents.

In this PoC [POC-IBN], a slice intent expresses a request for a network slice with two types of components: a set of top layer virtual functions, and a set of virtual switches and/or routers of L2/L3 VNFs. A service chain intent expressed a request for a service operated through a chain of service components running in L4-L7 virtual functions.

Following the intent classification methodology described step-by-step in Section 6.1. , we identify the following:

1. The Intent Solution is for the Data Center.

2. The Intent User Type is the Cloud Administrator for the slice intent and service chain intent.

3. The Type of Intent, is a Cloud Management intent, for the slice and service chain intent.

4. The Intent Scopes are connectivity and application.

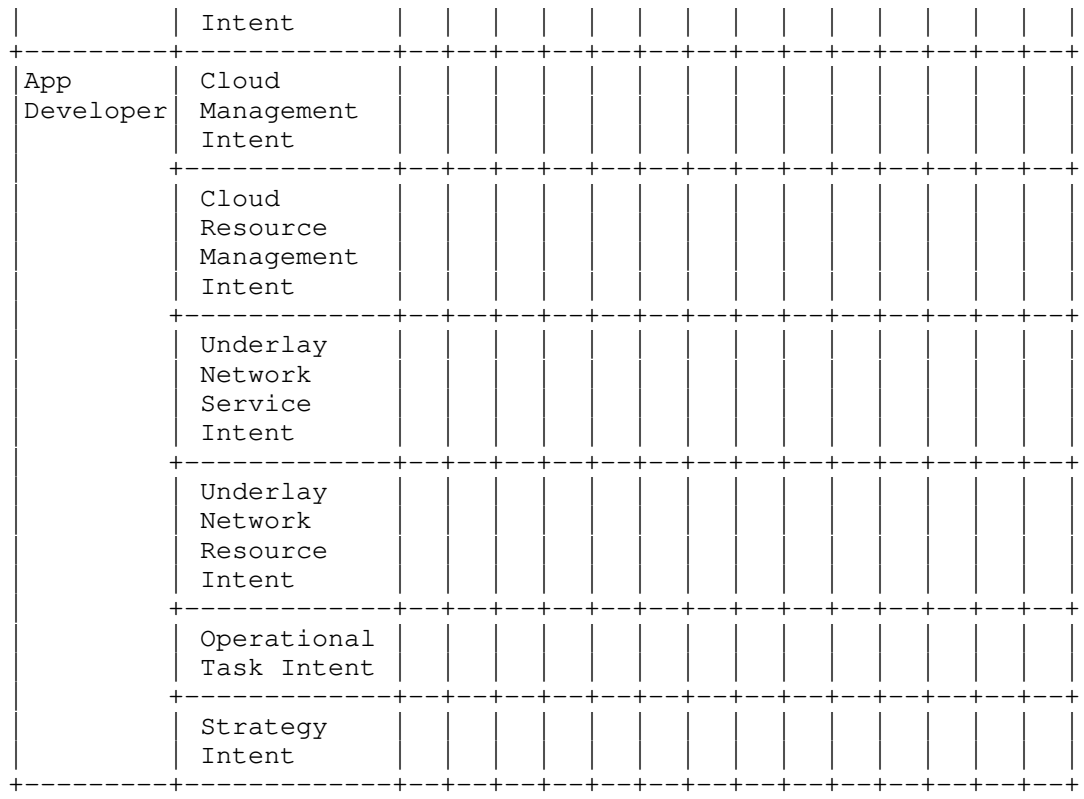
5. The Network Scope is a logical, and the resource scope is virtual.

6. The Abstractions are with technical feedback for the slice intent, and without technical feedback for the service chain intent

7. The life-cycle is persistent.

The following table shows how to represent this information in a tabular form, where the 'X' in the table refers to the slice and service chain intent.

Intent User	Intent Type	Intent Scope						DCN Res		DCN Net		ABS		L-C	
		C1	C2	C3	C4	C5	C6	C1	C2	C1	C2	C1	C2	C1	C2
Customer /Tenants	Customer Intent														
	Strategy Intent														
Cloud Admin	Cloud Management Intent	X		X				X		X		X	X	X	
	Cloud Resource Management Intent														
	Operational Task Intent														
	Strategy Intent														
Underlay Network Admin	Underlay Network Service Intent														
	Underlay Network Resource Intent														
	Operational Task Intent														
	Strategy														


Intent
App Developer
Cloud Management Intent
Cloud Resource Management Intent
Underlay Network Service Intent
Underlay Network Resource Intent
Operational Task Intent
Strategy Intent

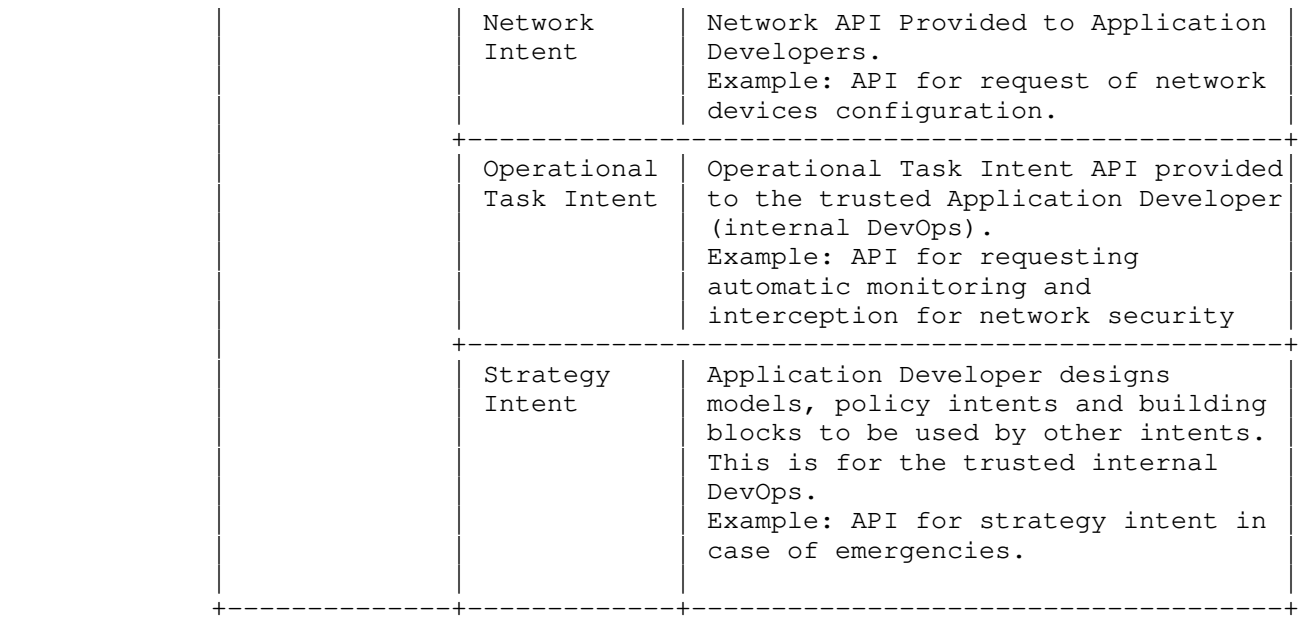
6.5. Intent Classification for Enterprise Solution

Users and Intent Types

The following table describes the Intent Users in Enterprise Solutions and their Intent Types.

Intent User	Intent Type	Intent Type Description
End-User	Customer Service Intent	Enterprise End-User Self-Service or Applications, Enterprise may have multiple types of End-Users. Example: Request access to VPN service. Request video conference between user A and B.
	Strategy Intent	This includes models and policy intents designed by End-Users to be used by End-User Intents and their Applications. Example: Create a video conference type for a weekly meeting.
Administrator (internal or MSP)	Network Service Intent	Service provided by the Administrator to the End-Users and their Applications. Example: For any user of application X, the arrival time of hologram objects of all the remote tele-presenters should be synchronised within 50ms to reach the destination viewer for each conversation session Create management VPN connectivity for type of service A. Operational statement: The job of the network layer is to ensure that the delay is between 50-70ms through the routing algorithm. At the same time,the node resources need to meet the bandwidth requirements of 4K

		video conferences.
	Network Intent	Administrator requires network wide configuration (e.g. underlay, campus) or resource configuration (switches, routers, policies). Example: Configure switches in campus network 1 to prioritise traffic of type A. Configure Youtube as business non-relevant.
	Operational Task Intent	Administrator requests execution of any automated task other than Network Service Intents and Network Intents. Example: Request network security automated tasks such as Web filtering and DDOS cloud protection.
	Strategy Intent	Administrator designs models, policy intents and workflows to be used by other intents. Automate any tasks that Administrator often performs. Example: In case of emergency, automatically shift all traffic of type A through network N.
Application Developer	End-User Intent	End-User Service / Application Intent API provided to the Application Developers. Example: API for request to open a VPN service.
	Network Service Intent	Network Service API Provided to Application Developers. Example: API for request network bandwidth and latency for hosting video conference.



Categories

The following are the proposed categories:
Intent Scope: C1=Connectivity, C2=Security, C3=Application, C4=QoS
Network (Net) Scope: C1=Campus, C2=Branch, C3=SD-WAN
Abstraction(ABS): C1=Technical(with technical feedback), C2=Non-technical (without technical feedback), see Section 5.2.
Life-cycle (L-C): C1=Persistent (Full life-cycle), C2=Transient (Short Lived)

The following is the Intent Classification Table Example for Enterprise Solutions.

Intent User	Intent Type	Intent Scope				Net			ABS		L-C	
		C1	C2	C3	C4	C1	C2	C3	C1	C2	C1	C2
End-User	End-User Intent											
	Strategy Intent											
Enterprise Administrator	Network Intent											
	Strategy Intent											
Application Developer	End-User Intent											
	Network Service Intent											
	Network Intent											
	Operational Task Intent											
	Strategy Intent											

7. Security Considerations

This document does not have any Security Considerations.

8. IANA Considerations

This document has no actions for IANA.

9. Contributors

The following people all contributed to creating this document, listed in alphabetical order:

Ying Chen, China Unicom
Richard Meade, Huawei
John Strassner, Huawei
Xueyuan Sun, China Telecom
Weiping Xu, Huawei

10. Acknowledgments

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Brian E Carpenter, Juergen Schoenwaelder, Laurent Ciavaglia, Xiaolin Song, Alexander Clemm, Daniel King, Mehdi Bezahaf, Yehia Elkhatib, Pedro Andres Aranda Gutierrez.

We thank to Walter Cerroni, Barbara Martini, Molka Gharbaoui for contributing with their 'A multi-level approach to IBN ' PoC demonstration a first attempt to adopt the intent classification methodology.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, June 2015.

[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)", March 2018.

[RFC3198] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J., Waldbusser, S., "Terminology for Intent-driven Management", RFC 3198, November 2001.

11.2. Informative References

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC7285] R. Alimi, R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, R. Woundy "Application-Layer Traffic Optimization (ALTO) Protocol", September 2014.

[ANIMA] Du, Z., "ANIMA Intent Policy and Format", 2017, <https://datatracker.ietf.org/doc/draft-du-anima-an-intent/>.

[ONF] ONF, "Intent Definition Principles", 2017, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-523_Intent_Definition_Principles.pdf>.

[ONOS] ONOS, "ONOS Intent Framework", 2017, <https://wiki.onosproject.org/display/ONOS/Intent+Framework/>.

[SUPA] Strassner, J., "Simplified Use of Policy Abstractions", 2017, <https://datatracker.ietf.org/doc/draft-ietf-supa-generic-policy-info-model/?include_text=1>.

[ANIMA-Prefix] Jiang, S., Du, Z., Carpenter, B., and Q. Sun, "Autonomic IPv6 Edge Prefix Management in Large-scale Networks", draft-ietf-anima-prefix-management-07 (work in progress), December 2017.

[TMF-auto] Aaron Richard Earl Boasman-Patel,et, A whitepaper of Autonomous Networks: Empowering Digital Transformation For the Telecoms Industry, inform.tmforum.org, 15 May, 2019.

[CLEMM] A. Clemm, L. Ciavaglia, L. Granville, J. Tantsura, "Intent-Based Networking - Concepts and Overview", Work in Progress, draft-clemm-nmrg-dist-intent-03, June 2020, https://tools.ietf.org/html/draft-irtf-nmrg-ibn-concepts-definitions-02

[POC-IBN] Walter Cerroni, Molka Gharbaoui, Barbara Martini, Davide Borsatti, "A multi-level approach to IBN", July 2020, https://www.ietf.org/proceedings/108/slides/slides-108-nmrg-ietf-108-hackathon-report-a-multi-level-approach-to-ibn-02

Authors' Addresses

Chen Li
China Telecom
No.118 Xizhimennei street, Xicheng District
Beijing 100035
P.R. China
Email: lichen.bri@chinatelecom.cn

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

Adriana Olariu
Huawei Technologies
Ireland
Email: adriana.olariu@huawei.com

Will(Shucheng) Liu
Huawei Technologies
P.R. China
Email: liushucheng@huawei.com

Pedro Martinez-Julia
NICT
Japan
Email: pedro@nict.go.jp

Jeferson Campos Nobre
Federal University of Rio Grande do Sul
Porto Alegre
Brazil
Email: jcnobre@inf.ufrgs.br

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain
Email: diego.r.lopez@telefonica.com

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

H. Yang
D. Chen
K. Yao
China Mobile

November 2, 2020

Network measurement intent
draft-yang-nmrg-network-measurement-intent-00

Abstract

This memo introduces network measurement intent, namely the process of realizing user or network operator to allocate network states as needed. And it can be as a specified user case of intent based network.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 May 6, 2021.

Copyright Notice

Table of Contents

1. Introduction

Since the development and implementation of intent based network (IBN) cannot be separated from accurate network state perception, accurate on demand network measurement technology is becoming more and more important. The combination of network measurement technology and IBN can achieve network performance acquisition based on user/network administrator intent-based, verify whether network measurement results meet the measurement intent, and further improve the accuracy of the configuration in IBN.

As the rise of IBN, different groups have different definitions of intent. For example, in [I-D.irtf-nmrg-ibn-concepts-definitions] defines intent as intent fulfillment and intent assurance. However, all different definitions of intent have some common characteristics, and can be classified according to [I-D.irtf-nmrg-ibn-intent-classification]. And in order to combine the network measurement intent with the existing drafts of IBN, we define the components of the network measurement intent processing process as follows:

- Intent Translation
- Intent Orchestration and pre-Verification
- Data Collection
- Intent Compliance Assessment

At the same time, according to [I-D.irtf-nmrg-ibn-concepts-definitions], network measurement intent can be classified as network intent, operational task intent or some other kinds of intent. And a detailed flow of network measurement intents will be given

2. Network Measurement Intent

Network measurement intent refers to the on-demand measurement of the network state based on the user/network operators' perceived intent of the network state. We will present the detailed process of it within each part and we will take the measurement of busy network performance as an example.

- Intent Translation. In this function, network measurement intents need to be translated into actions and requests taken against the network. In the measurement of busy network performances, due to dynamic changes such as daily network bandwidth occupancy rate, the period of network busy time is not fixed. As a result, Intent Translation can determine the threshold when the network state is busy on the same day based on the historical data learned by AI. And then determines the content to be measured.

- Intent Orchestration and pre-Verification. In this function, Intent Orchestration and pre-Verification determines the measurement scheme according to the required measurement content and equipment support degree, and verifies whether the measurement scheme is feasible. At the same time, it also needs to determine whether the network is busy according to the current network state. While the busy time threshold is exceeded, this function performs automatic network deployment, such as in CLI mode.

- Data Collection. This function is responsible for collecting data while determining the network is busy according to the current network state. And more importantly, this data collection process should start automatically.

- Intent Compliance Assessment. At the end, this function verifies whether the threshold meets the requirement and whether the network measurement intent is satisfied. If either of the two conditions is not satisfied, the network measurement intent will be modified and re-enter the Intent Orchestration and pre-Verification.

3. Summary

This memo introduces the network measurement intent, and give an example of network measurement of busy network performances. On the basis of existing intent drafts, this memo can be used as a use case for IBN.

4. Security Considerations

TBD.

5. IANA Considerations

This document has no requests to IANA.

6. References

6.1. 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>.

6.2. Informative References

[I-D.irtf-nmrg-ibn-concepts-definitions] Clemm, A., Ciavaglia, L., Granville, L., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", draft-irtf-nmrg-ibn-concepts-definitions-02 (work in progress), September 2020.

[I-D.irtf-nmrg-ibn-intent-classification] Li, C., Havel, O., LIU, W., Olariu, A., Martinez-Julia, P., Nobre, J., and D. Lopez, "Intent Classification", draft-irtf-nmrg-ibn-intent-classification-00 (work in progress), July 2020.

Authors' Addresses

Hongwei Yang
China Mobile
Beijing 100053
China

Email: yanghongwei@chinamobile.com

Danyang Chen
China Mobile
Beijing 100053
China

Email: chendanyang@chinamobile.com

Kehan Yao
China Mobile
Beijing 100053
China

Email: yaokehan@chinamobile.com

Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: May 20, 2021

C. Zhou
H. Yang
X. Duan
China Mobile
D. Lopez
A. Pastor
Telefonica I+D

November 16, 2020

Concepts of Digital Twin Network
draft-zhou-nmrg-digitaltwin-network-concepts-02

Abstract

Digital twin technology is becoming a hot technology in industry 4.0. The application of digital twin technology in network field helps to realize efficient and intelligent management and network innovation. This document presents an overview of the concepts of Digital Twin Network (DTN), provides the definition and DTN, and then describes the benefits and key challenges of DTN.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 May 20, 2021.

Table of Contents

1. Introduction

With the advent of 5G, Internet of Things and Cloud Computing, the scale of network is expanding constantly. Accordingly, the network operation and maintenance are becoming more complex due to higher complexity of network; and innovations on network will be more and more difficult due to the higher risk of network failure and higher trial cost.

Digital twin is the real-time representation of physical entities in the digital world. It has the characteristics of virtual-reality integration and real-time interaction, iterative operation and optimization, as well as full life-cycle, and full business data-

driven. At present, it has been successfully applied in the fields of intelligent manufacturing, smart city, complex system operation and maintenance [Tao2019].

A digital twin network platform can be built by applying digital twin technology to network and creating virtual image of physical network facilities. Through the real-time data interaction between physical network and twin network, the digital twin network platform can help the network to achieve more intelligent, efficient, safe and full life-cycle operation and maintenance.

2. Definition of Digital Twin Network

So far, there is no standard definition of digital twin network in networking industry or SDOs. This document attempts to define Digital Twin Network (DTN) as a virtual representation of the physical network, analyzing, diagnosing, simulating and controlling the physical network based on data, model and interface, so as to achieve the real-time interactive mapping between physical network and virtual twin network. According to the definition, DTN contains five key elements: data, mapping, model, interface and orchestration stack, as shown in Figure 1.

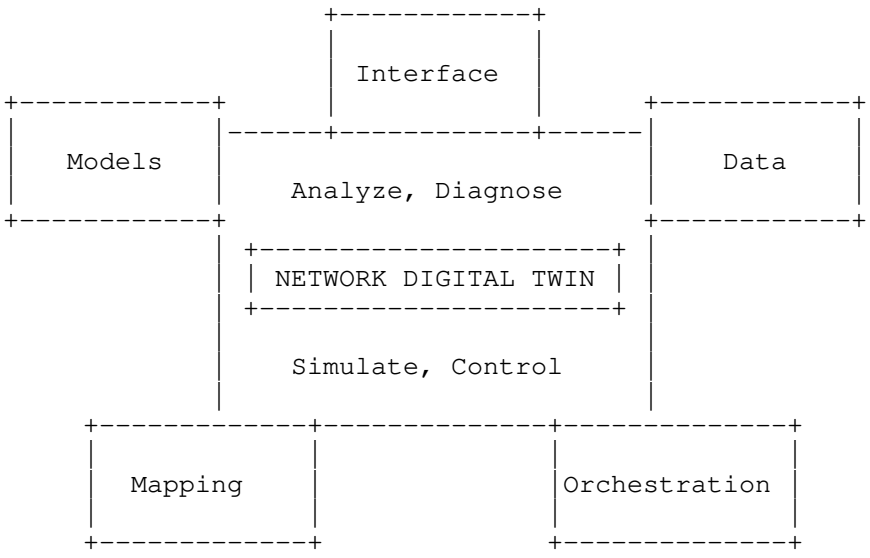


Figure 1: Key Elements of Digital Twin Network

- Data is cornerstone for constructing a DTN system, in which unified data repository can be the single source of the truth and provide timely and accurate data support.

- Real-time interactive mapping between physical network and virtual twin network is the most typical feature that DTN is different from network simulation system.

- Data model is the ability source of DTN. Various data models can be designed and flexibly combined to serve various network applications.

- Standardized interface is the key technique enabler, which can effectively ensure the compatibility and scalability of DTN system.

- The orchestration stack controls the flows of data and control actions. It relies on the dynamic lifecycle management of network models and elements to provide repeatablity (the capacity to replicate network conditions on demand) and reproducibility (the ability to replay successions of events, possibly under controlled variations).

3. Benefits of Digital Twin Network

DTN can help enable closed-loop network management across the entire lifecycle, from digital deployment and simulation, to visualized assessment, physical deployment, and continuous verification. In doing so, customers are able to achieve network-wide insights, precise planning, and rapid deployment in multiple areas, including networks, services, users, and applications. All the benefits of DTN can be categorized into three major types: low cost of network optimization, intelligent network decision making, and high efficient network innovation. The following sections describe the three types of benefits respectively.

3.1. Lower the cost of network optimization

With extremely large scale, network is becoming more and more complex and difficult to operate. Since there is no effective platform for simulation, traditional network optimization has to be tried on real network directly with long time cost and high service impact running on real network. This also greatly increases network operator's OpEX.

With DTN platform, network operators can well simulate the candidate optimization solutions before finally deploy them to real network. Compared with traditional methods, this is of quite low risk and will bring much less impact on real network. In addition, the operator's OpEX will be greatly decreased accordingly.

3.2. More intelligent for network decision making

Traditional network operation and management mainly focus on deploying and managing current services, while lacking of handling past data and predicting future status. This kind of passive and protective maintenance is difficult to adapt to large-scale network scenarios.

DTN can combine data acquisition, big data processing and AI modeling to achieve the assessment of current status, diagnosis of past problems, as well as prediction of future trends, then give the results of analysis, simulate various possibilities, and provide more comprehensive decision support. This will help network achieve predictive maintenance from current protective maintenance. The network behavioral repeatability and reproducibility properties in the DTN allow to evaluate different conditions and controlled variations of them, exploring choice as many times as needed to apply the better emulation and decision procedures.

3.3. High efficient for network innovation

Due to higher trial risk, real network environment is normally unavailable to network researcher when they explore innovation techniques. Instead, researchers have to use some offline simulation platforms. This greatly impacts the real effectiveness of the innovation, and greatly slow down the speed of network innovation. Moreover, risk-averse network operators naturally reluctant to try new technologies due to higher failure risk as well as the higher failure cost.

DTN can generate virtual twin entity of the real network. This helps researches explore network innovation (e.g. new network protocols, network AI/ML applications, etc.) efficiently, and helps network operators deploy new technologies quickly with lower risks. Take AI/ML application as example, it is a conflict between the continuous high reliability requirement (i.e. 99.999%) of network and the slow learning speed or phase-in learning steps of AI/ML algorithms. With DTN platform, AI/ML can fully complete the leaning and training with the sufficient data before deploy the model to the real network. This will greatly encourage more network AI innovations in future network.

Implementing Intent-Based Networking (IBN) via DTN can be another example to show how DTN improves the efficiency of deploying network innovation. IBN is an innovative technology for life-cycle network management. Future network will be possibly Intent-based, which means that users can input their abstract 'intent' to the network, instead of detailed policies or configurations on the network

devices. [I-D.irtf-nmrg-ibn-concepts-definitions] clarifies the concept of "Intent" and provides an overview of IBN functionalities. The key character of an IBN system is that user's intent can be assured automatically via continuously adjusting the policies and validating the real-time situation. To lower the impact on real network, several rounds of adjustment and validation can be simulated on the DTN platform instead of directly on physical netowrk. Therefore, DTN can be an important enabler platform to implement IBN system and speed up the deployment of IBN in customer's network.

3.4. Privacy and Regulatory Compliance

The requirements on data confidentiality and privacy on network service providers increase the complexity of network management, as intelligent decision engines depend on data flows. As a result, the improvement of data-enabled management requires complementary techniques providing strict control and security mechanisms to guarantee data privacy protection and regulatory compliance in these aspects. Some examples of these techniques can include payload inspection, including de-encryption user explicit consents, or data anonymization mechanisms.

Given DTN works with mapped traffic or services from real networks, but using traffic simulations, including automated tools for synthetic user activity. The lack of personal data permits to lower the privacy requirements and simplify privacy-preserving techniques, as the data is not coming from real users. As a result, DTN allows to focus on management improvements, without other concerns. Additionally, logging and auditing the DTN experiments and synthetic user activities provide additional information for further design and planning, without the need of traffic inspection.

3.5. Customize Network Operation Training

Networks architectures can be complex, and their operation and management require expert personnel and the learning curve can be steep in most cases. DTN offers an opportunity to train staff for customized networks and specific user needs. Several areas can benefit with the use of it. Two salient examples are the application of new network architectures and protocols, or the use of cyber-ranges to train security experts in threat detection and mitigation.

4. Reference Architecture of Digital Twin Network

So far, there is no reference or standard architecture for Digital Twin Network in network domain. Based on the definition of key elements of DTN described in section 2, reference architecture with

three layers of Digital Twin Network can be designed as below, shown in Figure 2.

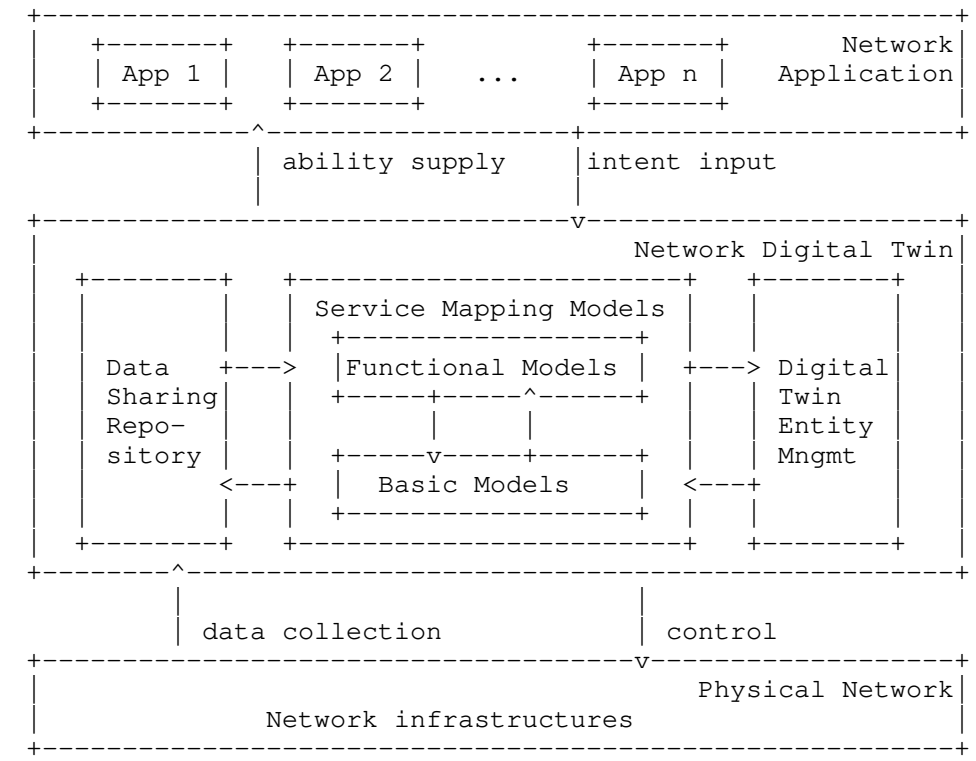


Figure 2: Reference Architecutre of Digital Twin Network

1. Bottom layer is Physical Network. All network elements in physical network exchange massive network data and control with network digital twin entity, via southbound interfaces. Physical network can be either telecommunication operator network, or data center network, campus network, industrial Internet of things or other network types.

2. Middle layer is Network Digital Twin Entity, which is the core of DTN system. This layer includes three key subsystems: Data Sharing Repository, Service Mapping Models and Digital Twin Entity Management.

   * Data Sharing Repository provides accurate and complete information for building various service models by collecting and updating the real-time operational data of various network elements through the southbound interface. In addition to

data storage, Data Sharing Repository is also responsible to provide data services for the Service Mapping Models sub-system, including fast retrieval, concurrent conflict, batch service, unified interface, etc.

* Service Mapping Models completes data-based modelling, provides data model instances for various network applications, and maximizes the agility and programmability of network services. The data models include two major types: basic models and functional models.

  + Basic Model refers to the network element model and network topology model of the network digital twin entity based on the basic configuration, environment information, operational state, link topology and other information of the network element, to complete the real-time accurate description of the physical network.

  + Functional model refers to various data models such as network analysis, simulation, diagnosis, prediction, assurance, etc. The functional models can be constructed and expanded by multiple dimensions: by network type, there can be models serving for single network domain or multi network domain; by function type, it can be divided into state monitoring, traffic analysis, security drill, fault diagnosis, quality assurance and other models; by generality, it can be divided into general model and special-purpose model. Specifically, multiple dimensions can be combined to create a data model for more specific application scenario.

* Digital Twin Entity Management completes the management function of digital twin network, records the life-cycle of the entity, visualizes and controls various elements of network digital twin, including topology management, model management and security management.

3. Top layer is Network Application. Various applications (e.g. Network intelligent O&M, IBN, etc.) can effectively run against Digital Twin Network platform to implement either conventional or innovative network operations, with low cost and less service impact on real network. Network application provide requirements to network digital twin entity via northbound interface; then the service is simulated by various service model instances; after fully verified, the change control can be deployed safely to physical network.

5. Challenges to build Digital Twin Network

As mentioned in above section, DTN can bring many benefits to network management as well as network innovation. However, it is still challenging to build an effective and efficient DTN system. The following are the major challenges and problems.

- Large scale challenge: The digital twin entity of large-scale network will significantly increase the complexity of data acquisition and storage, the design and implementation of model. And the requirements of software and hardware of the system will be very high.

- Compatibility issue: It is difficult to establish a unified digital twin platform with unified data model in the whole network domain due to the inconsistency of technical implementation and supporting functionalities of different manufacturers' devices in the network.

- Data modeling difficulties: Based on large-scale network data, data modeling should not only focus on ensuring the richness of model functions, but also need to consider the flexibility and scalability of the model. These requirements further increase the difficulty of building efficient and hierarchical functional data models.

- Real-time requirement: For services with high real-time requirements, the processing of model simulation and verification through DTN system will increase the service delay, so the function and process of the data model need to increase the processing mechanism under various network application scenarios; at the same time, the real-time requirements will further increase the system software and hardware performance requirements.

- Security risks: Network digital twin entity synchronizes all the data of physical network in real time, which will increase the security risk of user data, such as information leakage or more vulnerable to attack.

To solve the above problems and challenges, Digital Twin Network needs continuous optimization and breakthrough on key enabling technologies including data acquisition, data storage, data modeling, network visualization, interface standardization, and security assurance, so as to meet the requirements of compatibility, reliability, real-time and security under large-scale network.

6. Summary

The research and application of Digital Twin Network is just beginning. This document presents an overview of the concepts and definition of DTN. Looking forward, further researches on DTN usage scenarios, requirements, architecture and key enabling technologies should be promoted by the industry, so as to accelerate the implementation and deployment of DTN in real network.

7. Security Considerations

TBD.

8. IANA Considerations

This document has no requests to IANA.

9. References

9.1. 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>.

9.2. Informative References

[I-D.irtf-nmrg-ibn-concepts-definitions] Clemm, A., Ciavaglia, L., Granville, L., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", draft-irtf-nmrg-ibn-concepts-definitions-02 (work in progress), September 2020.

[Tao2019] Tao, F., Zhang, H., Liu, A., and A. Nee, "Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics, vol. 15, no. 4.", April 2019.

Authors' Addresses

Cheng Zhou
China Mobile
Beijing 100053
China

Email: zhouchengyjy@chinamobile.com

Hongwei Yang
China Mobile
Beijing 100053
China

Email: yanghongwei@chinamobile.com

Xiaodong Duan
China Mobile
Beijing 100053
China

Email: duanxiaodong@chinamobile.com

Diego Lopez
Telefonica I+D
Seville
Spain

Email: diego.r.lopez@telefonica.com

Antonio Pastor
Telefonica I+D
Madrid
Spain

Email: antonio.pastorperales@telefonica.com