ï»¿

Internet Research Task Force (IRTF)   A. Clemm
Request for Comments: 9315   Futurewei
Category: Informational   L. Ciavaglia
ISSN: 2070-1721   Nokia
                                                        L. Z. Granville
                     Federal University of Rio Grande do Sul (UFRGS)
                                                            J. Tantsura
                                                              Microsoft
                                                           October 2022

Intent-Based Networking - Concepts and Definitions

Abstract

   Intent and Intent-Based Networking are taking the industry by storm.
   At the same time, terms related to Intent-Based Networking are often
   used loosely and 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 the
   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 the foundation to guide further
   definition of associated research and engineering problems and their
   solutions.

   This document is a product of the IRTF Network Management Research
   Group (NMRG). It reflects the consensus of the research group,
   having received many detailed and positive reviews by research group
   participants. It is published for informational purposes.

Status of This Memo

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

   This document is a product of the Internet Research Task Force
   (IRTF). The IRTF publishes the results of Internet-related research
   and development activities. These results might not be suitable for
   deployment. This RFC represents the consensus of the Network
   Management Research Group of the Internet Research Task Force (IRTF).
   Documents approved for publication by the IRSG are not candidates for
   any level of Internet Standard; see Section 2 of RFC 7841.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   https://www.rfc-editor.org/info/rfc9315.

Copyright Notice

Table of Contents

1. Introduction

   This document is a product of the IRTF Network Management Research
   Group (NMRG). It reflects the consensus of the RG, receiving reviews
   and explicit support from many participants. It is published for
   informational purposes.

   In the past, interest regarding management and operations in the IETF
   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 [RFC3411] and the 200+
   MIBs that have been defined by the IETF over the years. More recent
   examples include YANG data model definitions [RFC7950] for aspects
   such as interface configuration, Access Control List (ACL)
   configuration, and Syslog configuration.

   There is a clear sense and reality that managing networks by
   configuring myriads of "nerd knobs" on a device-by-device basis is no
   longer an option in modern network environments. Significant
   challenges arise with keeping device configurations not only
   consistent across a network but also consistent with the needs of
   services and service features they are supposed to enable.
   Additional challenges arise with regard to being able to rapidly
   adapt the network as needed and to be able to do so at scale. 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.
   Among other things, this requires the ability to consume operational
   data, perform analytics, and dynamically take actions in a way that
   is aware of context as well as intended outcomes at near real-time
   speeds.

   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 of the
   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 input and operational guidance are commonly referred to as "intent," and a network that allows network operators to provide their input using intent is referred to as an "Intent-Based Network" (IBN), while a system that helps implement intent is referred to as an "Intent-Based System" (IBS). Those systems can manifest themselves in a number of ways -- for example, as a controller or management system that is implemented as an application that runs on a server or set of servers, or as a set of functions that are distributed across a network and that collectively perform their intent-based functionality.

However, intent is about more than just enabling a form of operator interaction with the network that involves higher-layer abstractions. It is also about the ability to let operators focus on what they want their desired outcomes to be while leaving details to the IBN (respectively IBS) about how those outcomes would be achieved. Focusing on the outcome enables much greater operational efficiency and flexibility at greater scale, in shorter time scales, and with less dependency on human activities (and therefore less possibility for mistakes). This also makes Intent-Based Networking an ideal candidate for artificial intelligence techniques that can bring about the next level of network automation [CLEMM20].

This vision has since caught on with the industry, 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 IBNs and IBSs (offering a full life cycle of intent), Software-Defined Network (SDN) controllers (offering a single point of control and administration for a network), and network management and Operations Support Systems (OSSs).

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 [M3010]. 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 concern. 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.

> As a note, in the context of this document, "users" generally refers to operators and administrators who are responsible for the management and operation of communication services and networking infrastructures, not to "end users" of communication services.

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 intent from related concepts. It also provides an overview of first-order principles of Intent-Based Networking as well as the 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 IBN-related research
problems is beyond the scope of this document. However, it should be
recognized that Intent-Based Networking has become an important topic
in the research community. Per IEEE Xplore [IEEEXPLORE], as of
December 2021, in the past decade since 2012, there have been 1138
papers with the index term "intent", of which 411 specifically
mention networking. The time period since 2020 alone accounts for
316 papers on intent and 153 for intent networking, indicating
accelerating interest. In addition, workshops dedicated to this
theme are beginning to appear, such as the IEEE International
Workshop on Intent-Based Networking [WIN21], as well as various
special journal issues [IEEE-TITS21]. A survey of current intent-
driven networking research has been published in [PANG20], listing
among the most pressing current research challenges aspects such as
intent translation and understanding, intent interfaces, and
security.

2. Definitions and Acronyms

ACL: Access Control List

API: Application Programming Interface

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.

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.

Intent-Based Management: The concept of performing management based
   on the concept of intent.

PBNM: Policy-Based Network Management

PDP: Policy Decision Point

PEP: Policy Enforcement Point

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

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.

Statement of Intent: A specification of one particular aspect or
   component of intent, also referred to as an intent statement.

SVoT: Single Version of Truth

User: In the context of this document, an operator and/or
   administrator responsible for the management and operation of
   communication services and networking infrastructure (as opposed
   to an end user of a communication service).

3. 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 policies (and Policy-Based
Network Management), and explains how they relate to intent and
Intent-Based Management.

3.1. Intent and Intent-Based Management

In this document, intent is defined as a set of operational goals
(that a network is supposed to meet) and outcomes (that a network is
supposed to deliver) defined in a declarative manner without
specifying how to achieve or implement them.

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 the 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 that differentiates intent
clearly from other types of policies needs to be introduced.

> One note regarding the use of the plural form of "intent": in
> the English language, the term "intent" is generally used only
> in singular form. However, the specification of intent as a
> whole usually involves the specification of several individual
> statements of intent, or intent statements. In some cases,
> intent statements are colloquially also referred to as
> "intents", although in general, the use of the term "intents"
> is discouraged.

Intent-Based Management aims to lead towards networks that are
fundamentally simpler to manage and operate, requiring only minimal
outside intervention. Networks, even when they are autonomic, are
not clairvoyant and have no way of automatically knowing particular
operational goals nor which 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 to the network by what informally
constitutes intent. That being said, the concept of intent is not
limited just to autonomic networks, such as networks that feature an
Autonomic Control Plane [RFC8994], but applies to any network.

Intent defines goals and outcomes in a manner that is purely
declarative, specifying what to accomplish, not how to achieve it.
Intent thus applies several important concepts simultaneously:

* It provides data abstraction: Users 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 do not need to be concerned even with how to
   achieve a given intent. What is specified is the desired outcome
   with the IBS automatically figuring out a course of action (e.g.,
   using an algorithm or applying a set of rules derived from the
   intent) for how to achieve the outcome.

The following are some examples of intent, expressed in natural
language for the sake of clarity (actual interfaces used to convey
intent may differ):

* "Steer networking traffic originating from endpoints in one
   geography away from a second geography, unless the destination
   lies in that second geography." (states what to achieve, not how.)

* "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." (states what to achieve, not how,
   providing additional guidance for how to trade off between
   different goals when necessary.)

* "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 historical mean." (a desired outcome, with a set of constraints for additional guidance, that does not specify how to achieve this.)

* "Ensure VPN services have path protection at all times for all paths." (a desired outcome of which it may not be clear how it can be precisely accommodated.)

* "Generate in situ Operations, Administration, and Maintenance (OAM) data and network telemetry for later offline analysis whenever significant fluctuations in latency across a path are observed." (goes beyond event-condition-action by not being specific about what constitutes "significant" and what specific data to collect.)

* "Route traffic in a Space Information Network in a way that minimizes dependency on stratospheric balloons unless the intended destination is an aircraft." (does not specify how to precisely achieve this; extrapolates on scenarios mentioned in [PANG20].)

* "For a smart city service, ensure traffic signal control traffic uses dedicated and redundant slices that avoid fate sharing." (a desired outcome with a set of constraints and additional guidance without specifying how to precisely achieve this; extrapolates on scenarios from [GHARBAOUI21].)

In contrast, the following are examples of what would not constitute intent (again, expressed in natural language for the sake of clarity):

* "Configure a given interface with an IP address." (This would be considered device configuration and fiddling with configuration knobs, not intent.)

* "When interface utilization exceeds a specific threshold, emit an alert." (This would be a rule that can help support network automation, but a simple rule is not an intent.)

* "Configure a VPN with a tunnel from A to B over path P." (This would be considered as a configuration of a service.)

* "Deny traffic to prefix P1 unless it is traffic from prefix P2." (This would be an example of an access policy or a firewall rule, not intent.)

In networks, in particular in networks that are deemed autonomic, intent should ideally be rendered by the network itself, i.e., translated into device-specific rules and courses of action. Ideally, intent would not need to 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 abstractions. 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. The system providing this point acts as the operational front end for the network through which users can direct requests at the network and from which they can receive updates about the network. It may appear to users as a single system, even if it is implemented in a distributed manner. In turn, it interacts with and manages other systems in the network as needed in order to realize (i.e., to fulfill and to assure) the desired intent. Likewise, the vast majority of network devices may be intent-agnostic and focus only (for example) on the actual forwarding of packets. Many devices may also be constrained in terms of their processing resources. This means that not every device may be able

to act on intent on its own. Again, intent in those cases can be
achieved by a separate system that performs the required actions.

Another reason to provide intent functionality from a conceptually
centralized point is in cases where the realization of a certain type
of intent benefits from global knowledge of a network and its state.
In many cases, such a global view may be impractical to maintain by
individual devices, for example due to the volume of data and time
lags that are involved. It may even be impractical for devices to
simply access such a view from another remote system if such were
available.

All of this implies that in many cases, certain intent functionality
needs to be provided by functions that are specialized for that
purpose and that may be provided by dedicated systems (which in some
cases could also co-host other networking functions). For example,
the translation of specific types of intent into corresponding
courses of action and algorithms to achieve the desired outcomes may
need to be provided by such specialized functions. Of course, to
avoid single points of failure, the implementation and hosting of
such functions may still be distributed even if conceptually
centralized.

Regardless of its particular implementation in a centralized or
decentralized manner, an IBN is a network that can be managed using
intent. This means that it is able to recognize and ingest intent of
an operator or user and configure and adapt itself 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 IBN
will be able to figure out on its own how to achieve the outcome.
Similarly, an IBS is a system that allows users to manage a network
using intent. Such a system will serve as a point of interaction
with users and implement the functionality that is necessary to
achieve the intended outcomes, interacting for that purpose with the
network as required.

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 that is presented here is more
encompassing and ambitious, as 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 abstraction at the network level
is more limited and would not necessarily qualify as intent.
Likewise, ingestion and recognition of intent may not necessarily
occur via an API based on function invocations and simple request-
response interactions but may involve other types of human-machine
interactions such as dialogs to provide clarifications and
refinements to requests.

3.2. Related Concepts

3.2.1. 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 and 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, and 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 [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 of
communication services who might be billed for the services provided.

Instantiating a service typically involves multiple aspects:

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

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

* 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 of a communication service with the service).

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

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

The realization of service models involves a system, such as a
controller, that provides provisioning logic. This includes breaking
down high-level service 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 and validates whether
the new changes have been accepted and the new operational/derived
states are achieved and in sync with the intent/desired state. In
addition to instantiating and creating new instances of a service,
updating, modifying, and decommissioning services also need to be
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 and 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 (composition), while troubleshooted bottom up
(decomposition). 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.

[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 an IBS. Instead,
the management of service models requires the development of
sophisticated algorithms and control logic by network providers or

system integrators.

3.2.2. 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 specify 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.
This is notwithstanding the fact that policy-based management may
make use of the concept of abstractions (such as, "Bob gets gold
service") that hide from the user the specifics of how that
abstraction is rendered in a particular deployment.

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 decoupled 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, data
center (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 that is responsible for
deploying policies across the network and monitoring 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 action. Unlike intent, the
definition of those rules (and courses of action) 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 PBNM. In that
sense, policy constitutes a lower level of abstraction than intent,
and it is conceivable for IBSs to generate policies that are
subsequently deployed by a PBNM system, allowing PBNM to support
Intent-Based Networking.

3.2.3. 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, generally transcends
individual devices, and 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.

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 the 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. Policies let users define what to do under
    what circumstances, but they do 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 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 IBS. 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-based systems
and IBSs 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 policies 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 requiring large

data sets; explanations of actions that the system actually takes
provide a different set of challenges. Analogous to IBSs, users
focus on what they would like the learning system to accomplish but
not how to do it.

4. 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 of Truth (SVoT).
       The SSoT is an essential component of an IBS 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/desired state
       and actual/operational states of the system and determining drift
       between them. SSoT and the drift information provide the basis
       for corrective actions. If the IBS is equipped with the means to
       predict states, it can further develop strategies to anticipate,
       plan, and pro-actively act on any 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 in order 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 querying, polling, or
       filtering measured and correlated information in order to create
       so-called "views." These views can serve the users of the IBS.
       In order to create intent statements as single sources of truth,
       the IBS must follow well-specified and well-documented processes
       and models. In other contexts, SSoT is also referred to as the
       invariance of the intent [LENROW15].

   2. One touch but not one shot. In an ideal IBS, the user expresses
       intent 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 IBS has the capabilities
       or means to recognize intentions in any form of data. However,
       the zero- or one-touch approach should not distract from 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 IBS could be designed as an
       interactive and iterative process. Depending on the level of
       abstraction, the intent expressions may initially contain more or
       less implicit parts and imprecise or unknown parameters and
       constraints. The role of the IBS 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 means of the system itself. In
       addition, the IBS 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 IBS 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 intent using the user's own terms, by supporting
       different forms of expression for individual statements of intent
       and being capable of refining the intent expressions to well-
       formed and exploitable expressions. The dual principle of
       autonomy and supervision allows operating a system that will have
       the necessary levels of autonomy to conduct its tasks and
       operations without requiring the 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 IBS is a learning system. In contrast to an
    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 IBS, the user
    only declares what the system is supposed to 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 IBS of capabilities or means for learning,
    reasoning, and knowledge representation and management. The
    learning abilities of an IBS can apply to different tasks such as
    optimization of the intent rendering or intent refinement
    processes. The fact that an IBS 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 IBS to anticipate or forecast future
    system state and response to changes in intent 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.

5. Capability exposure. Capability exposure consists in the need
    for expressive network capabilities, requirements, and
    constraints to be able to compose/decompose intent 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 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, or 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, and 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 IBSs and their supporting architecture. Accordingly, they
need to be considered when deriving functional and operational
requirements.

5. Intent-Based Networking - Functionality

    Intent-Based Networking involves a wide variety of functions that 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 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 is not
   meant to be transactional, i.e., "set and forget", but is expected
   to remain in effect over time (unless explicitly stated
   otherwise). 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.

5.1. 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.

5.1.1. 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 IBS.
Typically, those functions go beyond those provided by a non-intent-
based API, although non-intent-based APIs may also still be provided
(and needed for interactions beyond human users, i.e., with other
machines). Many cases would also involve a set of intuitive and
easy-to-navigate workflows that guide users through the intent
ingestion phase, making sure that all inputs that are necessary for
intent modeling and consecutive translation have been gathered. They
may support unconventional human-machine interactions, in which a
human will not simply give commands but instead a human-machine
dialog is used to provide clarifications, to explain ramifications
and trade-offs, and to facilitate refinements.

The goal is ultimately to make IBSs as easy and natural to use and
interact with as possible, in particular allowing human users to
interact with the IBS in ways that do not involve a steep learning
curve that forces the user to learn the "language" of the system.
Ideally, it will be the IBSs that are increasingly able to learn how
to understand the user, as opposed to the other way around. Of
course, further research will be required to make this a reality.

5.1.2. Intent Translation

A second set of functions needs to translate user intent into courses
of action and requests to take against the network, which will be
meaningful to network configuration and provisioning systems. These
functions lie at the core of IBS, bridging the gap between
interaction with users on the one hand and the 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 and 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 will need
to be configured across the network. Heuristics and algorithms to do
so may evolve over time to optimize outcomes that may depend on a
myriad of dynamic network conditions and context.

5.1.3. 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.

5.2. Intent Assurance

Intent 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.

5.2.1. 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, and generating and collecting telemetry data.
Monitoring and observation are required as the 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.

5.2.2. 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 and is held by
SSoT. 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 a control
plane or lower-level management operations that inadvertently cause
behavior changes that conflict with intent that was orchestrated
earlier. IBSs and Networks need to be able to detect when such drift
occurs or is about to occur as well as assess the severity of the
drift.

5.2.3. 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.

5.2.4. Abstraction, Aggregation, Reporting

The outcome of Intent Assurance needs to be reported back to the user
in ways that allow 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 human users.

6. 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 5 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 IBS with the human user. It involves the functions that allow users to articulate and the IBS 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 outcomes and 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 the network operations infrastructure. This includes the functions used to translate an intent into a course of action as well as the algorithms that are used to plan and optimize those courses of action also in consideration of feedback and observations from the network. It also includes the functions to analyze and aggregate observations from the network in order to validate compliance with the intent and to take corrective actions as necessary. In addition, it includes functions that abstract observations from the network in ways that relate them to the intent as communicated by users. This facilitates the reporting functions in the user space.

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

```
          User Space    :      Translation / IBS     :  Network Ops
                         :            Space           :     Space
                         :                            :
                         :                            :
         +----------+    : +----------+   +-----------+ : +-----------+
Fulfill  |recognize/|--->| |translate/|-->|   learn/  |-->| configure/|
         |generate  |    | |          |   |   plan/   |   | provision |
         |intent    |<---| |  refine  |   |   render  | : |           |
         +----^-----+    : +----------+   +-----^-----+ : +-----------+
              |          :                      |       :        |
              |          :                      |       :        |
         .....|...................................|.................|.....
              |          :                +---+---+   :        v
              |          :                |validate|   :   +----------+
              |          :                +----^---+ <----| monitor/  |
Assure   +---+---+       : +----------+   +-----+---+   :   | observe/  |
         |report |   <---| |abstract  |<---|  analyze | <----|           |
         +-------+       : +----------+   |aggregate |   :   +----------+
                         :                +---------+   :
```
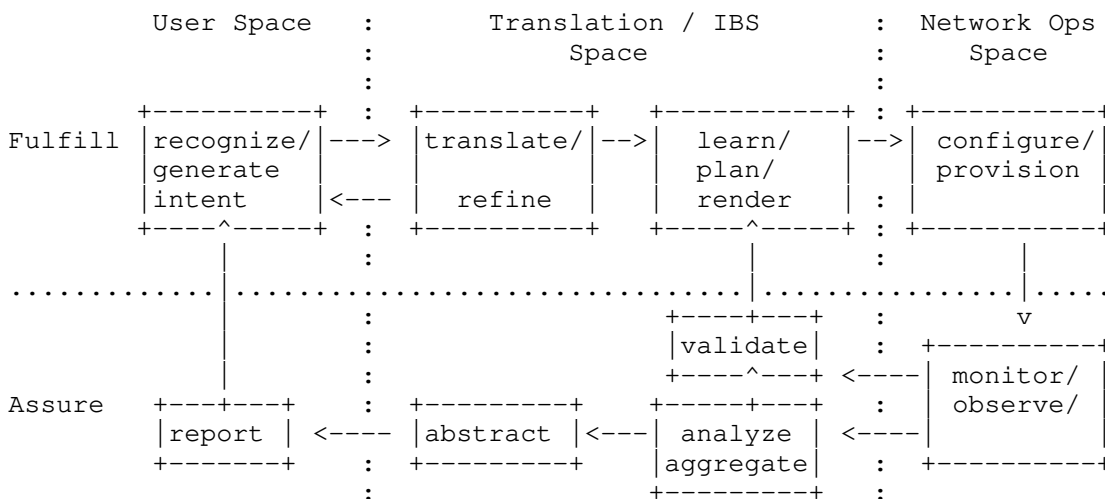
                     Figure 1: Intent Life Cycle

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 represents closed-loop automation that 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 loop addresses and counteracts any intent drift that
    may be occurring, using observations to assess the degree of the
    network's intent compliance and automatically prompting actions to
    address any discrepancies. Likewise, the loop allows to assess
    the effectiveness of any actions that are taken in order to
    continuously learn and improve how intent needs to be rendered in
    order to achieve the desired outcomes.

*   The "outer" intent control loop extends to the user space. It
    includes the user taking action and adjusting their intent based
    on observations and feedback from the IBS. Intent is thus
    subjected to a life cycle: It comes into being, may undergo
    refinements, modifications, and changes of time, and may at some
    point in time even get retracted.

7. Additional Considerations

   Given the popularity of the term "intent," it is tempting to broaden
   its use to encompass other related concepts, resulting in "intent-
   washing" that paints those concepts in a new light by simply applying
   new intent terminology to them. A common example concerns referring
   to the northbound interface of SDN controllers as "intent interface."
   However, in some cases, this actually makes sense not just as a
   marketing ploy but as a way to better relate previously existing and
   new concepts.

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

   Operational Intent: defines intent related to operational goals of
      an operator; it 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.

8. IANA Considerations

   This document has no IANA actions.

9. 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 IBS itself.

*   Mitigating the effects of erroneous, harmful, or compromised
    intent statements.

*   Expressing security policies or security-related parameters with
    intent statements.

Securing the IBS aims at making the IBS operationally secure by
implementing security mechanisms and applying security best
practices. In the context of Intent-Based Networking, such
mechanisms and practices may consist of intent verification and
validation, operations on intent by authenticated and authorized
users only, and protection against or detection of tampered
statements of intent. 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-offs should be considered as well.

Mitigating the effects of erroneous or compromised statements of
intent aims at making the IBS operationally safe by providing
checkpoint and safeguard mechanisms and operating principles. In the
context of Intent-Based Networking, such mechanisms and principles
may consist of the ability to automatically detect unintended,
detrimental, or abnormal behavior; the ability to automatically (and
gracefully) roll back or fall back to a previous "safe" state; the
ability to prevent or contain error amplification (due to the
combination of a higher degree of automation and the intrinsic higher
degree of freedom, ambiguity, and implicit information conveyed by
intent statements); and 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 intent
statements may inadvertently propagate and compromise security. In
addition, compromised intent statements (for example, 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 as intent
consists of 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, and of isolation in processing functions;

*   level(s) of encryption; and

*   authorized entities for given operations.

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 a 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.

10. Informative References

[BOUTABA07]
          Boutaba, R. and I. Aib, "Policy-Based Management: A
          Historical Perspective", Journal of Network and Systems
          Management (JNSM), Vol. 15, Issue 4,
          DOI 10.1007/s10922-007-9083-8, November 2007,
          <https://doi.org/10.1007/s10922-007-9083-8>.

[CLEMM20] Clemm, A., Faten Zhani, M., and R. Boutaba, "Network
          Management 2030: Operations and Control of Network 2030
          Services", Journal of Network and Systems Management
          (JNSM), Vol. 28, Issue 4, DOI 10.1007/s10922-020-09517-0,
          October 2020,
          <https://doi.org/10.1007/s10922-020-09517-0>.

[GHARBAOUI21]
          Gharbaoui, M., Martini, B., and P. Castoldi,
          "Implementation of an Intent Layer for SDN-enabled and
          QoS-Aware Network Slicing", 2021 IEEE 7th International
          Conference on Network Softwarization (NetSoft), pp. 17-23,
          DOI 10.1109/NetSoft51509.2021.9492643, June 2021,
          <https://doi.org/10.1109/NetSoft51509.2021.9492643>.

[IEEE-TITS21]
          Garg, S., Guizani, M., Liang, Y-C., Granelli, F., Prasad,
          N., and R. R. V. Prasad, "Guest Editorial Special Issue on
          Intent-Based Networking for 5G-Envisioned Internet of
          Connected Vehicles", IEEE Transactions on Intelligent
          Transportation Systems, Vol. 22, Issue 8, pp. 5009-5017,
          DOI 10.1109/TITS.2021.3101259, August 2021,
          <https://doi.org/10.1109/TITS.2021.3101259>.

[IEEEXPLORE]
          IEEE, "IEEE Xplore", <https://ieeexplore.ieee.org/>.

[LENROW15] Lenrow, D., "Intent As The Common Interface to Network
          Resources", Intent Based Network Summit 2015 ONF Boulder:
          IntentNBI, February 2015.

[M3010]   ITU-T, "Principles for a telecommunications management
          network", ITU-T Recommendation M.3010, February 2000.

[NETWORK-SLICE]
          Farrel, A., Ed., Drake, J., Ed., Rokui, R., Homma, S.,
          Makhijani, K., Contreras, L. M., and J. Tantsura,
          "Framework for IETF Network Slices", Work in Progress,
          Internet-Draft, draft-ietf-teas-ietf-network-slices-14, 3
          August 2022, <https://datatracker.ietf.org/doc/html/draft-
          ietf-teas-ietf-network-slices-14>.

[PANG20]   Pang, L., Yang, C., Chen, D., Song, Y., and M. Guizan, "A
          Survey on Intent-Driven Networks", IEEE Access, Vol. 8,
          pp.22862-22873, DOI 10.1109/ACCESS.2020.2969208, January
          2020, <https://doi.org/10.1109/ACCESS.2020.2969208>.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An
          Architecture for Describing Simple Network Management
          Protocol (SNMP) Management Frameworks", STD 62, RFC 3411,
          DOI 10.17487/RFC3411, December 2002,
          <https://www.rfc-editor.org/info/rfc3411>.

[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>.

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

[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>.

[RFC8994]   Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An
            Autonomic Control Plane (ACP)", RFC 8994,
            DOI 10.17487/RFC8994, May 2021,
            <https://www.rfc-editor.org/info/rfc8994>.

[SERVICE-MAPPING-YANG]
            Lee, Y., Ed., Dhody, Dhruv., Ed., Fioccola, G., Wu, Q.,
            Ed., Ceccarelli, D., and J. Tantsura, "Traffic Engineering
            (TE) and Service Mapping YANG Data Model", Work in
            Progress, Internet-Draft, draft-ietf-teas-te-service-
            mapping-yang-11, 11 July 2022,
            <https://datatracker.ietf.org/doc/html/draft-ietf-teas-te-
            service-mapping-yang-11>.

[SLOMAN94] Sloman, M., "Policy Driven Management for Distributed
            Systems", Journal of Network and Systems Management
            (JNSM), Vol. 2, Issue 4, pp. 333-360, December 1994.

[STRASSNER03]
            Strassner, J., "Policy-Based Network Management", August
            2003.

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

[WIN21]   Ciavaglia, L. and E. Renault, "1st International Workshop
            on Intent-Based Networking", IEEE International Conference
            on Network Softwarization, June 2021,
            <https://netsoft2021.ieee-netsoft.org/program/workshops/
            win-2021/>.

Acknowledgments

Authors' Addresses

   Alexander Clemm
   Futurewei
   2330 Central Expressway
   Santa Clara, CA 95050
   United States of America
   Email: ludwig@clemm.org

   Laurent Ciavaglia
   Nokia

Route de Villejust
91620 Nozay
France
Email: laurent.ciavaglia@nokia.com

Lisandro Zambenedetti Granville
Federal University of Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves
Porto Alegre-RS
9500
Brazil
Email: granville@inf.ufrgs.br

Jeff Tantsura
Microsoft
Email: jefftant.ietf@gmail.com