Internet Research Task Force
Internet-Draft
Intended status: Informational
Expires: 7 May 2026

K. Yao, Ed.
D. Chen, Ed.
China Mobile
J. Jeong, Ed.
Sungkyunkwan University
Q. Wu
Huawei
C. Yang
Xidian University
L. Contreras
Telefonica
G. Fioccola
Huawei

3 November 2025

Use Cases and Practices for Intent-Based Networking
draft-irtf-nmrg-ibn-usecases-02

Abstract

This document proposes several use cases of Intent-Based Networking (IBN) and the methodologies to differ each use case by following the lifecycle of a real IBN system. It includes the initial system awareness and data collection for the IBN system and the construction of the IBN system, which consists of intent translation, deployment, verification, evaluation, and optimization. Practice learning and general learning are also summarized to instruct the construction of next generation network management systems with the integration of IBN techniques. Finally, this document discusses three aspects for the deployment of IBN systems on the real world. They are Multi-Domain Deployment, Network Digital Twin, and IBN with Artificial Intelligence (AI).

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.

This Internet-Draft will expire on 7 May 2026.

Copyright Notice

Table of Contents

1. Introduction

[RFC9315] gives the concepts and definition of Intent-Based Networking (IBN), and [RFC9316] proposes a comprehensive taxonomy of the intent classifications. Although the intent life cycle has been defined, including all the core functional components like intent injestion, intent translation, policy generation, and intent assurance, there is still a big gap between defining these high-level functionality and building realistic IBN systems. This document summarizes the methodologies, proposes several IBN use cases, and then practice learning and general learning when building an IBN system. Main objectives of this document is to instruct future research directions of IBN and other related network management technologies in the perspective of network operators and vendors as well as service providers.

2. Methodologies for Building IBN Systems

This section summarizes the methodologies to build an IBN system. These methodologies refer to the modeling of an intent life cycle and its high-level core functional components, as well as the specific solutions to implement those components [RFC9315]. The methodologies are essential to build a real IBN system, beyond the definition in [RFC9315]. The methodologies to an IBN system are composed of several important parts, including the system awareness and data collection, the construction of an IBN system, integration and deployment, evaluation, optimization, and the reconfiguration of intents and policies.

2.1. System Awareness and Data Collection

System awareness requires the collection of various network status indicators, like network traffic and resources. Building a valuable dataset is essential for IBN systems. A comprehensive data collection depends on suitable methods and tools, appropriate sampling metrics, and reasonable granularity for data collection.

1. Methods and Tools

   * There are many existing ways to collect network data which can be primarily classified into two types, active measurement and passive measurement. Active measurement like In-band Network Telemetry (INT) [INT] can grab networking information by inserting timestamps into the programmable field of on-path packets. Passive measurement, on the other hand, uses some tools like Tcpdump or Wireshark to collect data at specific targets, like endpoint servers. IBN systems need both of the ways to collect data, depending on what scenarios they might be applied to.

2. Metrics

   * Metrics include traffic-related and network-related information. Traffic-related metrics include performance indicators, such as latency, throughput, and traffic congestion signals. Network-related information includes network device information, such as the number and health status of ports, and network topology information (e.g., link connectivity and structures). To meet a specific user intention, such as load balancing and congestion elimination on the entire network, IBN systems need to collect and process traffic and device related information.

3. Granularity

   * Network Traffic: Network traffic is usually collected in various forms, such as per-packet [INT] and per-flow (or per-flowlet) [IntFlow], and these are two most typical types of data collection. Per-packet tracking lets each packet be tracked, which is very accurate, but it requires greater monitoring overhead and state maintenance overhead [INT]. In contrast, per-flow tracking does not need to maintain too many states, and it generally uses five-tuples (i.e., source IP address, destination IP address, source port number, destination port number, transport layer protocol) to identify each flow, which often brings good observation results [IntFlow]. Other collection methods are like per-cell and

per-flow [IntFlow]. Per-cell tracking tracks each cell unit whose length remains unchangeable, which is more friendly to system management and control. This method is often applied to Artificial Intelligence (AI) data center network monitoring. Per-flowlet tracking cuts a flow into several small flows at a certain interval, which is more suitable for implementing refined load balancing scenarios. Thus, the IBN system should select an appropriate traffic collection granularity (e.g., packet, flow, flowlet, and cell).

* Time Granularity: Time granularity means that the data acquisition needs to adopt the appropriate time interval for data sampling. In the extreme case, data is collected without interruption. For example, the status information of each data packet is reported to a monitoring module without interruption. This collection method often brings too much redundant information, which leads to a lot of storage and computing overhead to the monitoring module. However, the method of sampling without interruption or at a very low time interval can better observe micro-bursts of the networking system. A micro-burst occurs when a large amount of burst data is received in milliseconds. For some black-box network systems and some high-concurrency network systems, it is necessary to sacrifice a certain amount of storage and computing costs to collect data in a finer granularity time slot, so as to make better a trade-off between system overhead and data acquisition accuracy. By analyzing the historical behavior of IBN systems, a reasonable time interval can be selected for data acquisition.

* Spatial Granularity: Spatial granularity indicates that it is necessary to select an appropriate physical scope of a network for data collection. In some cases, the information collection method based on the whole network and the whole domains may not be suitable for all situations, and sometimes the results obtained from the processing and analysis of the collected data may not be accurate (e.g., RTT-based congestion control in data center networking) or incur too much overhead (e.g., hop-by-hop performance monitoring over the Internet). The best way is to match the most appropriate spatial granularity for user intents. For example, in wide-area data transmission, users need to select an optimal path. In this case, sampling is not required for all paths from a source to a destination. Only partial sampling is required for certain path segments which share endpoints, to ensure the correctness of decision makings on path setup in a scenario of multi-path data transmission.

2.2. The Construction of an IBN System

An IBN system consists of intent translation module, policy generation and mapping module, intent verification module, intent deployment module, monitoring module, intent validation module, and policy optimization module. Each module in the IBN system matches with each module in the Intent Life Cycle in [RFC9315]. The different construction methods and different construction tools used in these modules may affect the advantages of realizing a user intent. For different modules, we summarize the methods and tools that have been used and may be used.

1. Intent Translation

   * Translating and refining intents require the system to explore and exploit the semantic relationships of different service intents [I-D.gu-nmrg-intent-translator][I-D.pedro-ite]. It is necessary to build a general model to extract the key semantic information from the service intents in different representation forms. In the intent translation module, several possible intent expressions and translation methods are as follows:

     - A limited range of templates are preset in advance, and users can only express corresponding intentions by filling in or selecting templates. The advantage of this method is that the requirements for users and translation are very low, and all users can use it without learning. The disadvantage is that there are many restrictions, which can only be achieved through a preset template, but the preset template is limited, and cannot really meet the flexible and diverse needs of users.

     - Using Natural Language Processing (NLP), such as Flan-T5 [Flan-T5] and GPT-3 [GPT-3], for intent translation is another possible approach. NLP is used to convert a user’s intent in a human language (e.g., English) into a text intent in a computer programming language (e.g., XML, JSON, and YAML). This translation from a verbal intent in a natural language to a text intent in a computer programming language is performed by an intent translator [I-D.gu-nmrg-intent-translator]. The advantage of this method is high flexibility, users can directly express their intents in a natural language according to their own needs, without being limited by templates. The disadvantage is that it is difficult to implement and has high requirements for the intent translation module. This needs to be able to accurately identify the real intent of

a user, and different intent expression paradigms will affect the generation of subsequent policies. Thus, it is necessary to formalize normative intent expression grammars.

- In addition, there are some preset expression languages for IBN networks, such as Nile (Network Intent Language) [Nile] and NEMO (Network Modeling Language) [I-D.xia-sdnrg-nemo-language]. In the designs of these languages' expressions, most of them consider the flexibility of the expressions, which can be extended and adjusted according to the intent scenario of the business under consideration. However, these language designs have some disadvantages (e.g., the capabilities of intent expressions). Most of the users are network practitioners, requiring the users to have certain network knowledge background.

2. Policy Translation

* In an Intent-Based Network, the translation from a user intent to the corresponding network policy is required. The generated network policy needs to be mapped to an appropriate network function or network device to execute the policy. Thus, both the policy translation and mapping are required for intent enforcement in the target intent-based network.

* A given user intent needs to consider both the intent and the network state, that is, the policy needs to satisfy the user intent and ensure that a network operation can be executed to satisfy the requested intent. The policy generation module can be implemented by setting up a repository of "intent" and "policy", and mapping relationship between the intent and policy should be stored and updated as knowledge in a knowledge datastore (e.g., knowledge graph [Knowledge-Graph]) according to various intents and dynamic network state telemetry.

* There is a mapping submodule in the policy translation module. This mapping submodule can select an appropriate network function or network device to execute the requested policy. The selection of such a network function or network device can be done by a set-cover algorithm or decision tree algorithm. One of these selection algorithms searches for a network function or network device that can accommodate the keywords in the policy.

* Similar to different ways of expressing an intent, there are different approaches for the policy generation.

  - As opposed to the default template-based representation in the intent translation module, the simplest approach to policy generation is based on a default template or rule-based provisioning. After the user completes the corresponding intent expression through the graphical interface (e.g., a web-based graphical user interface (GUI)), a user or an AI agent can select the corresponding policy according to the preset template in the policy generation or the rules in a constructed rule-based policy generator. Similar to the above analysis, this approach has the advantage of being very simple to implement, but the disadvantage is that it is too restrictive and only a limited number of preset strategies can be selected.

  - The second common method of policy translation is inference-based generation, such as reasoning based on keywords in an intent expression, associating keywords with policies, and using Circular Reasoning [Circular-Reasoning] to generate policies. This method is more flexible than the template class description method, but the precision of policy generation is more related to the keyword extraction, and there is some uncertainty. In addition, there are policy generation methods based on network service description, which are widely used in Service Function Chaining (SFC) [RFC7665], network slicing or Network Functions Virtualization (NFV) [ETSI-NFV][ETSI-NFV-Release-2]. In essence, this approach can also be seen as inference-based strategy generation.

  - In addition to the above methods, AI technology-based policy generation methods have also emerged in recent years, such as machine learning technology, which selects the corresponding policies through model training according to keywords extracted from an intent expression. With the development of AI technology, in addition to selecting preset policies, for example, based on Deep Reinforcement Learning [DRL], reasonable reward functions are set to generate strategies that consider user intents and network status.

3. Policy Verification

* Policy verification checks whether the policy meets a specific user’s requirements or not. Also, it includes policy conflict detection and policy conflict resolution [AI-Intent-Network].

- The policy conflict detection includes two types: the conflicts between different policies themselves and the conflicts between policies and network states of the target network to perform the requested policy. The conflict of the policies may be due to the conflict between the network states that different users want to obtain. The simplest example is that both users A and B request to increase the bandwidth of 10Gbps, but the network bandwidth of the shared network for users A and B is less than 20Gbps. This conflict caused by different user requirements can be resolved by a policy conflict handler that checks whether the policies can be deployed in practice, that is, you can choose to execute only the policies that can be executed according to the preset rules, and reject other conflicting policies. If the generated policy conflicts with the network state, the intent-based system must detect that the generated policy cannot be executed by the target network. Also if the generated policy cannot be executed, the policy needs to be re-generated. Otherwise, the failed policy generation should be reported to the intent user as a failure.

- In terms of whether the policy is satisfied or not, the first way is to feedback the result to the user, and the user judges whether it is satisfied or not. For this purpose, the execution result can be presented through a graphical user interface. The second way is to use an AI agent such as deep reinforcement learning [DRL] to determine whether the results meet the needs or not.

4. Policy Deployment

* Policy deployment is to deploy the policy translated from an intent into a network function or network device in a target network and let the configurations or commands of the policy operate in the network.

  - The policy translator delivers a policy with detailed configurations or commands to a policy renderer which deploys the policy into target network functions or devices (e.g., switch, router, firewall, web filter, and DDoS-attack mitigator), which are called target network entities.

  - The policy renderer delivers the policy to the target network entities with a policy delivery protocol such as NETCONF [RFC6241], RESTCONF [RFC8040], or REST API [REST].

- The target network entities execute their own configuration for the requested network services which are specified by the policy.

5. Policy Monitoring

* Policy monitoring is to collect monitoring data from network entities (e.g., switch, router, firewall, web filter, and DDoS-attack mitigator) for policy validation to judge whether the requested policy is enforced well or not in the target network.

  - Network entities send their monitoring data to a validation module (e.g., analyzer) via a delivery protocol such as NETCONF [RFC6241], RESTCONF [RFC8040], and REST API [REST].

  - The validation module stores the monitoring data into its local repository for further analysis and investigation.

6. Policy Validation

* Policy validation is to judge whether the requested policy is satisfied by network entities in a target network or not. The policy may have goals in terms of performance (e.g., throughput, delay, and loss rate) and services (e.g., firewall, web filter, and DDoS-attak mitigator).

  - A validation module (e.g., analyzer) uses the collected monitoring data for evaluation and check whether the required goals for each policy are met with specific metrics from the monitoring data or not. This checking can be performed by Artificial Intelligence (AI) and Machine Learning (ML) algorithms.

  - Evaluation results need to be delivered to an optimization module (e.g., optimizer) which can augment the existing policy or generate a new policy for further improvement.

7. Policy Optimization

* Policy optimization is to augment the existing policy or generate a new policy to meet the goals of the requested intent. With the evaluation results, an optimization entity (e.g., optimizer) performs optimization for each registered intent.

- There are two kinds of optimization, such as Quality of Service (QoS) and Service Provisioning. First, the optimizer for QoS deals with the improvement of performance metrics (e.g., throughput, delay, and loss rate). Second, the optimizer for service provisioning handles the service requirements (e.g., firewall filtering, web filtering, and DDoS-attack mitigation). For each optimization, the optimizer augments the existing policy or generates a new policy for further improvement. It delivers the policy to the policy renderer so that the renderer can enforce the augmented or generated policy into the target network entities.

- Thus, the steps from Policy Deployment to Policy Optimization construct a closed-loop policy control to guarantee the goals of the requested intent in a target network. This is network service automation using the IBN technology.

8. Intent Report

* Intent report is to abstract and report the operation results in a target network for a given intent. Abstration submodule abstracts results in the form of text, figures, and tables. Reporting submodule delivers the abstracted results to the user to let him (or her) know the activity and performance of the network.

  - There are two kinds of submodules for intent report, such as abstraction submodule and reporting submodule.

  - The abstraction submodule analyzes the activity and performance of target network entities in the target network. The analysis is expressed in the form of text, tables, and figures by various statistics, AI, ML, and graphics tools.

  - The reporting submodule delivers the analysis report to the user (e.g., network administrator and operator) so that (s)he may check the enforcement and quality of the requested network services for the given user intent in the target network with relevant network entities. The user can render another intent or modified intent to satisfy his (or her) user intent in the target network.

2.3. Mapping between IBN System and Intent Life Cycle

There is a mapping between the modules of an IBN System in Section 2.2 and the modules of the Intent Life Cycle in [RFC9315].

* Intent Translation in the IBN System is mapped to (i) Intent Ingestion and Interaction with Users and (ii) Intent Translation in the Intent Life Cycle.

* Policy Translation in the IBN System is mapped to Intent Orchestration in the Intent Life Cycle.

* Policy Verification in the IBN System is mapped to Intent Orchestration in the Intent Life Cycle.

* Policy Deployment in the IBN System is mapped to Intent Orchestration in the Intent Life Cycle.

* Policy Monitoring in the IBN System is mapped to Monitoring in the Intent Life Cycle.

* Policy Validation in the IBN System is mapped to Intent Compliance Assessment in the Intent Life Cycle.

* Policy Optimization in the IBN System is mapped to Intent Compliance Actions in the Intent Life Cycle.

* Intent Report in the IBN System is mapped to Abstraction, Aggregation, and Reporting in the Intent Life Cycle.

3. IBN Use Cases

In this section, we will describe several scenarios where IBN can be applied. These use cases can reflect the aforementioned methodologies of IBN systems from different perspectives.

3.1. IBN for Routing and Path Selection

IBN can be applied in building network path and generating routing policies according to network administrators’ requests.

3.1.1. IBN for Service Function Chaining

An intent-based dynamic SFC is an example to solve the network management challenges (e.g., cross-domain orchestration and service functions are tightly coupled with the underlying equipment). An Intent-Based Network Management (IBNM) platform can be developed on top of the OpenStack [OpenStack]. The system architecture is shown as Figure 1, which includes the application layer, the intent-enabled layer and the infrastructure layer. The application layer collects intents from various users and applications, and provides a number of programmable network management services to the users. The intent-enabled layer consists of the intent translation module, intelligent policy mapping module, and intent guarantee module, whose functions are to build a bridge between the application layer and the infrastructure layer. Heterogeneous physical devices are deployed in the infrastructure layer. This layer can execute management instructions from the intent-enabled layer and upload underlying network situation information to the intent-enabled layer. Information interaction between different layers is done through different interfaces, such as the northbound and southbound interfaces.

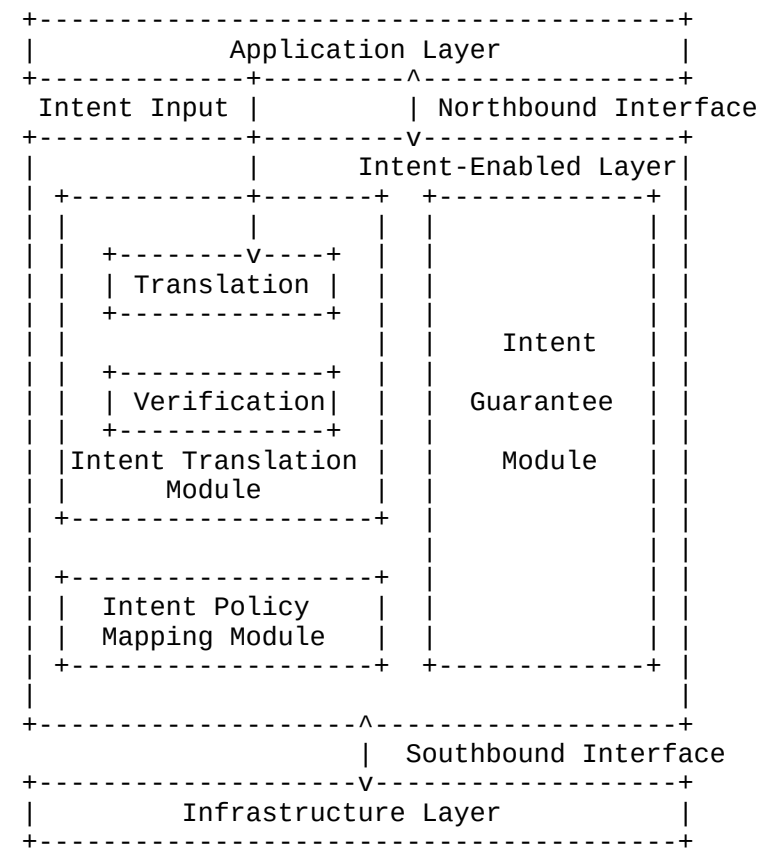

Figure 1: The Architecture of an IBNM System

The system demonstration implements the whole process from intent input to intent translation to intent policy generation for intent deployment, and the details are as follows.

The user inputs an intention that is cross-domain link-building requests in a natural language at a web page. An exemplary intent is "Transfer a common-level video service from user A in Beijing to user B in Nanjing while constraining the execution time of the intent."

With the intention in the natural language, the intent translation module outputs a conflict-free translation result (e.g., intent), which indicates that the external intent input (called intention) and the intent translation module have communicated with each other. The translation result is intent tuples, which are displayed on the front-end interface (e.g., web interface) in the form of name-value pairs. After the intent translation module, the translation result will be converted to a JavaScript Object Notation (JSON) request (e.g., intent) and transmitted to the intent policy mapping module.

The intent policy mapping module translates the JSON request as an intent into policies as an SFC: service function 1 (e.g., network address translation) and service function 2 (e.g., firewall). It then constructs the SFC request (having name, tenant_id, description, service requirements, etc.). Then it queries whether there is an atomic policy combination that satisfies the current intent requirements in the policy repository or not.

Following that, an SFC is constructed based on the SFC interface, which is extended by Neutron. OpenStack schedules network resources, constructs subnets and ports, and generates a two-dimensional space topology. Meanwhile, during the SFC construction process, the intent guarantee module monitors and manages network resource utilization as well as network failures in real time.

Overall, IBNM achieves the decoupling of service application and network, and cross-domain network orchestration, while reducing the complexity of network management.

3.1.2. IBN for SRv6 Networks

For the automation of configuration and monitoring of Segment Routing version six (SRv6) routers, an IBN-based SRv6 network management is proposed by [I-D.park-nmrg-ibn-network-management-srv6]. The proposed IBNM framework for SRv6 consists of system components and interfaces, as shown in Figure 2. This figure shows an IBNM framework for 5G core networks using SRv6. This framework is built on the framework for Interface to Network Security Functions (I2NSF) [RFC8329].

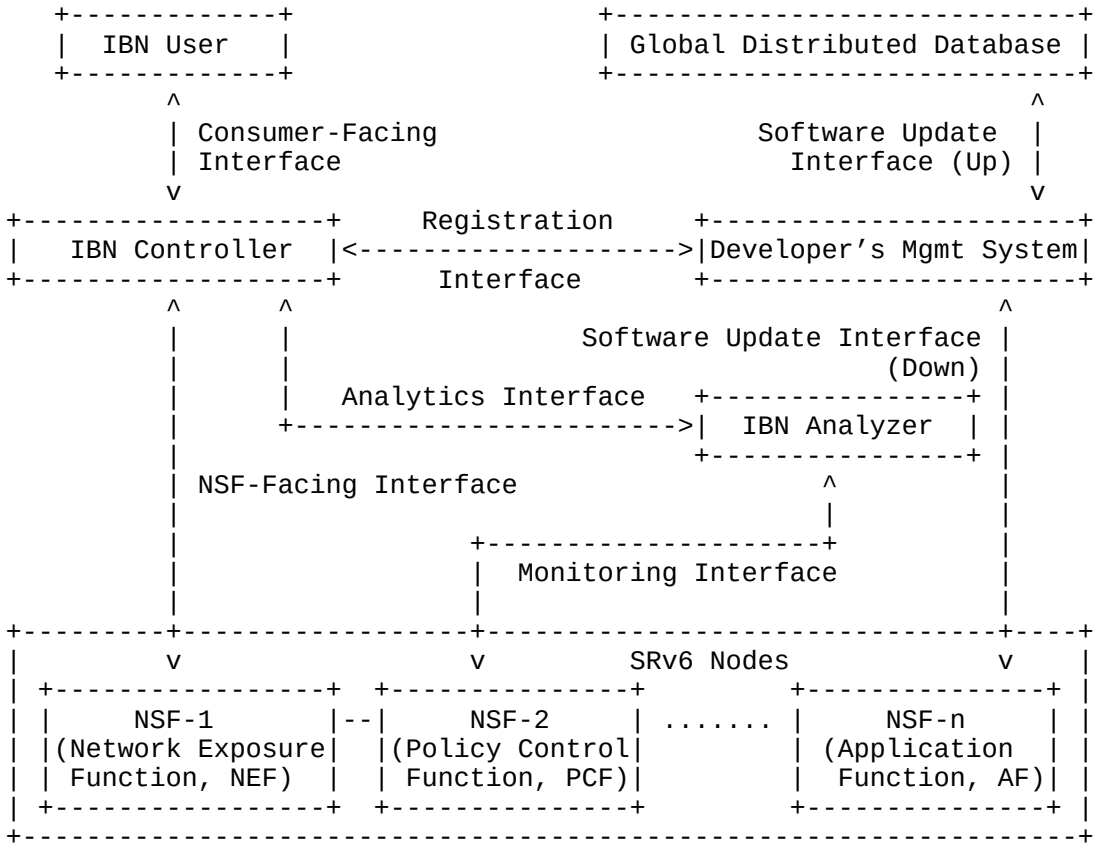

Figure 2: Intent-Based Network Management in SRv6 Networks

A high-level network policy for SRv6 nodes (e.g., NSFs) is constructed by a Consumer-Facing Interface YANG data model. On the other hand, a low-level network policy is constructed by an NSF-Facing Interface YANG data model. A high-level network policy is delivered to IBN Controller by IBN User via the Consumer-Facing Interface. On the other hand, a low-level network policy is delivered to a Network Service Function (NSF) by IBN Controller via the NSF-Facing Interface.

To automate Network Policy Translation (NPT), IBN Controller needs a network policy translator performing the translation of a high-level network policy into the corresponding low-level network policy (i.e., SRv6 policy [RFC9256]). As a prerequisite step for this automatic NPT service, the IBN framework needs to associate a high-level YANG data model and a low-level YANG data model in an automatic manner, like a data model mapper [I-D.ietf-spring-sr-policy-yang], [SPT].

For the policy assurance, NSFs send their monitoring data to IBN Analyzer on the basis of either periods or events via Monitoring Interface. IBN Analyzer analyzes the NSF monitoring data by AI and ML algorithms to check whether NSFs are working appropriately according to a network policy (called an intent). IBN Analyzer sends a report with either policy reconfiguration or feedback to IBN Controller for further actions for the policy assurance. Optionally, IBN Controller sends a report to IBN User to report the network status and events for the IBN User's high-level policy (called intent).

3.2. IBN for Guaranteeing Service-Level Agreement

The performance metrics for Service-Level Agreement (SLA) in a target network are packet loss, delay, throughput, etc. An IBN-based approach can ensure that these performance parameters comply with well-defined SLAs.

If we consider the delay, the simple schematic diagram is shown in Figure 3. Different thresholds, warning values, and alert values should be set for network delay measurement in advance. When the delay value is below warning, the network is normal and the business is normal. When the delay is between a warning value and an alert value, the network fluctuation is abnormal, but the business is normal. When the delay exceeds the alert value, both the network and business are abnormal. For the delay in different thresholds, different measurement strategies should be adopted:

* When the network delay exceeds an alert value, or when the historical data predicts that the delay will exceed the alert value, passive measurement requires 100% sampling of business data, and the transmission frequency of active measurement is adjusted to the maximum value. At the same time, the log and alarm data of the whole network equipment is collected to realize the most fine-grained measurement of the network, locate the root cause of the problem, and repair the network in time.

* When the network delay exceeds a warning value but is lower than an alert value, passive measurement samples 60% of business data, and the transmission message frequency of the active measurement is adjusted to the median value, and the running state data of some key devices in the network is collected synchronously.

* When the network delay is less than a warning value, passive measurement data is sampled at 20%, and active measurement message frequency is adjusted to the lowest value, and the network equipment running state of key nodes can be collected as needed.

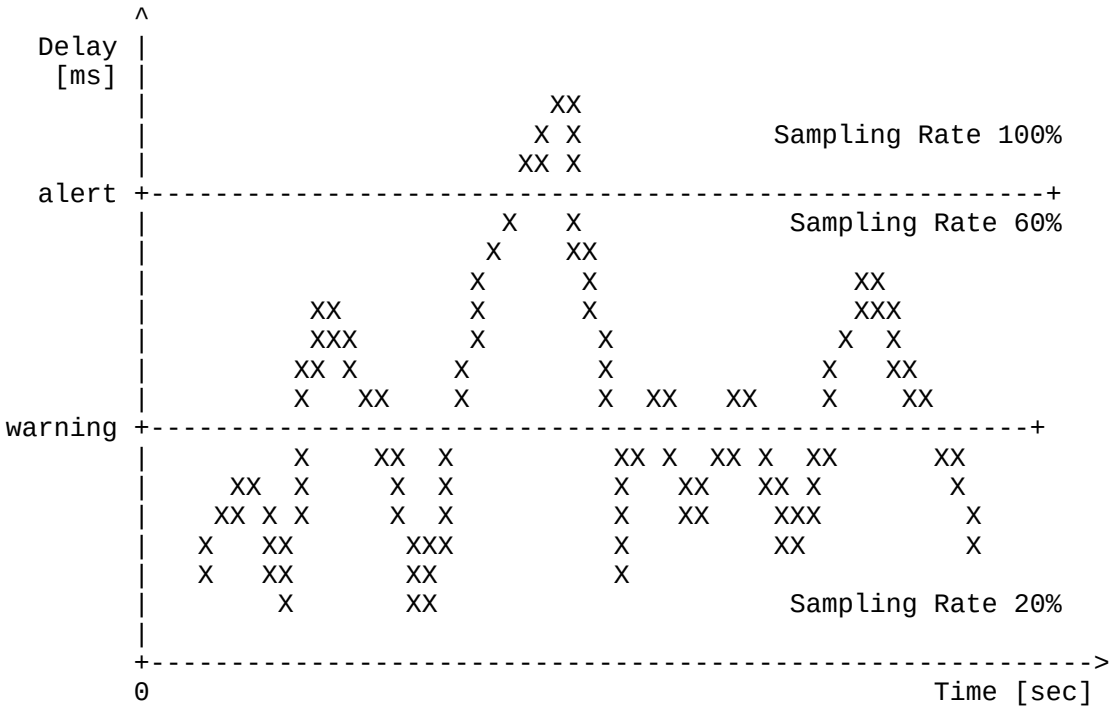

Figure 3: Network SLA Performance Metrics

The desired approach is to accurately measure the network state, especially when there are some issues affecting the service, but at the same time, reduce the resources to be employed to achieve the desired accuracy.

The example of network delay has been provided, but the same approach can be applied to other performance indicators (e.g., packet loss, throughput, and goodput) as well.

3.2.1. On-Path Telemetry Methods

The On-path Telemetry Methods refers to performance measurement techniques that can provide flow information on the entire forwarding path on a per-packet basis in real time. Differently from the traditional active tools for Operations, Administration, and Maintenance (OAM), which inject test packets for measurements, the On-path Telemetry Methods (e.g., AltMark [RFC9341] and IOAM [RFC9197]) allow to monitor real service packets and thereby allowing to directly measure network performance indicators from the live networks.

Alternate-Marking Method [RFC9341] (AltMark) and In-situ Operations, Administration, and Maintenance (IOAM) [RFC9197] are the standard On-path Telemetry Methods. AltMark is a technique used to perform packet loss, delay, and jitter measurements by marking in-flight packets according to the methodology described in [RFC9341] and [RFC9342]. IOAM is a method that allows to produce operational and telemetry information that may be exported using the in-band or out-of-band method. The data types and data formats for IOAM data records have been defined in [RFC9197] and [RFC9326].

With AltMark and IOAM, the real-time traffic monitoring of the network can be used to optimize the network performance. Figure 4 shows an exemplary traffic monitoring system with a high-level IBN workflow for dynamic network control based on traffic monitoring with On-path Telemetry Methods.

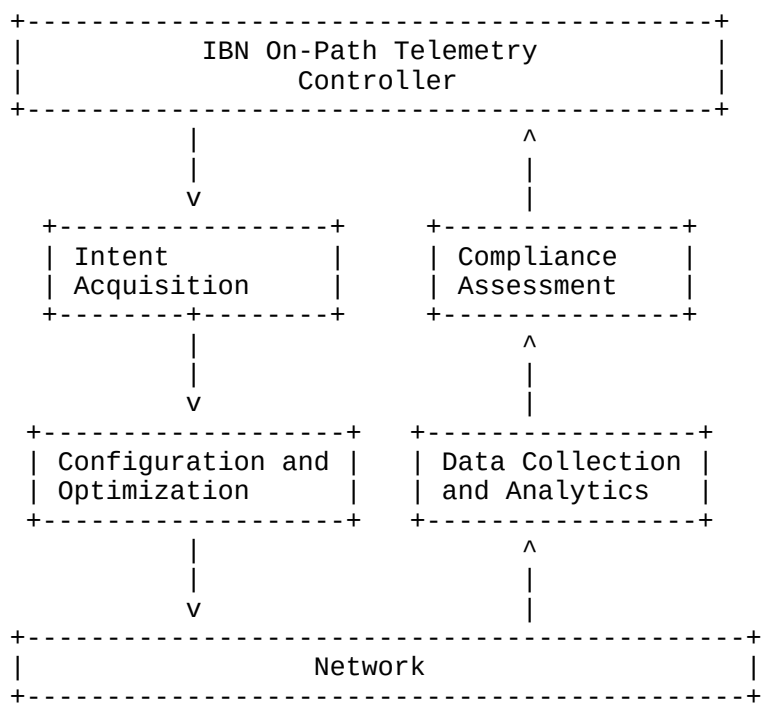

Figure 4: A Traffic Monitoring System with IBN-Based On-Path Telemetry

The Controller for IBN On-Path Telemetry in Figure 4 configures the monitoring of the network according to a specific performance measurement intent. For this monitoring, AltMark or IOAM can be used. Then it collects data and analytics from the selected

methodology (e.g., AltMArk or IOAM) in order to verify the compliance with the intent. Optimization actions can be eventually taken and can be related to network path modification or performance measurements variation.

The next section describes an example of the workflow for IBN-based On-path Telemetry focusing on the use of [RFC9342].

3.2.1.1. Clustered Alternate-Marking Methodology

The Clustered Alternate-Marking framework in [RFC9342] adds flexibility to performance measurements in [RFC9341], because it can reduce the order of magnitude of the packet counters. This allows the Controller to supervise, control, and manage AltMark measurements in large-scale networks.

[RFC9342] introduces the concept of cluster partition of a network. The monitored network can be considered as a whole or split into clusters that are the smallest subnetworks (e.g., group-to-group segments), maintaining the packet loss property for each subnetwork. The clusters can be combined in new connected subnetworks at different levels, which can form new clusters, depending on the level of detail to achieve.

A clustered performance measurement intent represents the spatial accuracy, that is, the size of the subnetworks to consider for the monitoring. It is possible to start the monitoring without examining in depth and, in case of necessity, the “network zooming” approach can be used.

This approach is called “network zooming” and can be performed in two different ways:

1. To change the traffic filter and select more detailed flows;

2. To activate new measurement points by defining more specified clusters.

The network-zooming approach implies that some filters, rules or flow identifiers are changed. But these changes must be done in a way that do not affect the performance. Therefore, there could be a transient time to wait until the new network configuration takes effect. Anyway, if the performance issue is relevant, it is likely to last for a time much longer than the transient time.

The concrete steps of the clustered performance measurement intent are as follows:

* The performance measurement intent acquisition is initially recognized. For example, the intent can be a specific SLA for the network in terms of performance parameter values. Then, the performance measurement intent is analyzed and it is translated into specific configurations and measurement policy, such as network partition and the spatial accuracy needed for the monitoring.

* The configuration step arranges and calibrates the measurement with the specific configuration according to the measurement policy in order to split the whole network into clusters at different levels. Note that, for the configuration, the YANG Data Model for the Alternate Marking Method [I-D.ydt-ippm-alt-mark-yang] can be used.

* The data collection and analytics step gets the measurement data from the different clusters, and then validates the actual performance for each cluster against the required performance according to the intent. Note that, for the collection of the measurement data, the On-path Telemetry YANG Data Model [I-D.fz-ippm-on-path-telemetry-yang] or the IPFIX Alternate-Marking Information [I-D.ietf-opsawg-ipfix-alt-mark] can be used.

* The compliance assessment checks whether the initial intent is met or not, that is, for example, if a cluster is experiencing a packet loss or the delay is higher than the expected value. In this case, the Controller is notified of such an outage in order to modify the cluster partition of the network for further investigation. The network configuration can be immediately modified in order to perform a new partition of the network but only for the cluster with bad performance. In this way, the problem can be localized with successive approximation up to a flow detailed analysis. This is the so-called “Intent-Based Closed-Loop Performance Management”.

3.3. IBN for Cloud-Based Security Service Management

A Cloud-Based Security Service Management is proposed in [I-D.jeong-i2nsf-security-management-automation]. It describes Security Management Automation (SMA) of cloud-based security services in the framework of Interface to Network Security Functions (I2NSF) [RFC8329]. The security management automation deals with closed-loop security control, security policy translation, and security audit. To support these three features in SMA, an augmented architecture of the I2NSF framework is proposed by introducing new system components and new interfaces.

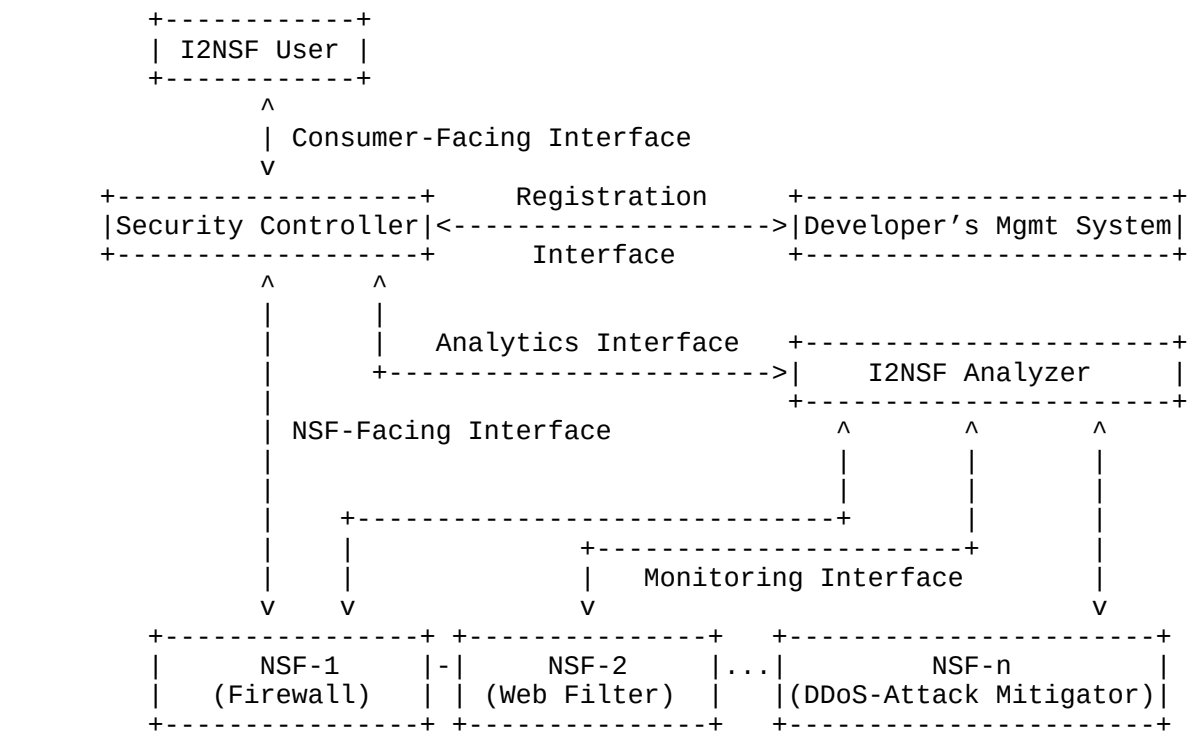

Figure 5: Security Management Automation in I2NSF Framework

Figure 5 shows an IBN-driven I2NSF framework for Security Management Automation (called SMA) of cloud-based security service management. I2NSF User composes a high-level security policy (as an intent) and delivers it to Security Controller. Security Controller translates the high-level security policy into the corresponding low-level security policy that is understandable to Network Security Functions (NSFs) for actual security services. Security Controller has a Security Policy Translator (SPT) for this security policy translation [SPT].

As shown in Figure 5, for closed-loop security control, this I2NSF framework has Monitoring Interface and Analytics Interface along with I2NSF Analyzer. I2NSF Analyzer collects monitoring data from NSFs via Monitoring Interface. It analyzes the monitoring data using Artificial Intelligence (AI) and Machine Learning (ML). I2NSF Analyzers delivers a policy reconfiguration message (e.g., defense against a new security attack) or feedback information message (e.g., action for handling overloaded computing and communication resources) to Security Controller. Security Controller receives the message and takes an appropriate action for the message, such as translating the message into a security policy reconfiguration for target NSFs and taking a remedy action for the feedback information.

Therefore, with a security policy translator and a closed-loop security control, we can provide service customers with IBN-based security services according to the intent life cycle in [RFC9315].

3.4. IBN for IoT Device Management in 5G Networks

A Network Management Automation (NMA) can be provided for cellular network services in 5G networks [I-D.jeong-nmrg-ibn-network-management-automation]. This NMA is feasible on top of an IBN-empowered framework. It deals with a closed-loop network control, network intent translator, and network management audit. To support these three features in NMA, it specifies an architectural framework with system components and interfaces. Also, this framework can support the use cases of NMA in 5G networks such as the data aggregation of Internet of Things (IoT) devices, network slicing, and the Quality of Service (QoS) in Vehicle-to-Everything (V2X).

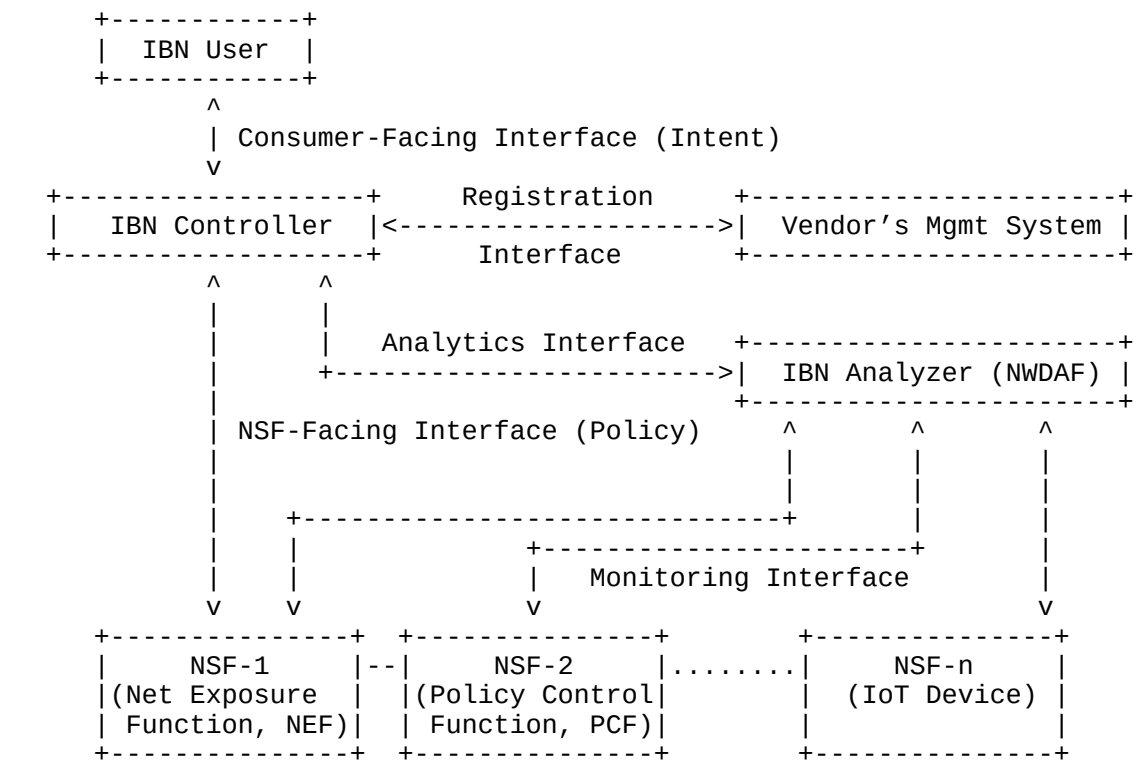

Figure 6: Network Management Automation in IBN Framework for 5G Networks

Figure 6 shows an IBN framework for Network Management Automation in 5G networks. This framework is based on the I2NSF framework for cloud-based security services [RFC8329][I-D.jeong-i2nsf-security-management-automation]. Like the framework for Security Management Automation (called SMA) of cloud-based security services, this framework supports an intent translation with a Network Intent Translator (NIT) and a closed-loop control mechanism, it realizes an IBN-based IoT device management in 5G networks.

An intent is expressed with YAML [YAML] according to an intent specification in [TS-28-312]. The delivery protocol of an intent and a tranlsated policy can be REST API [REST].

3.5. IBN for Sofware-Defined Vehicle Management

Software-Defined Vehicle (SDV) is an electrical vehicle with a software platform (e.g., AUTOSAR [AUTOSAR], Eclipse SDV [Eclipse-SDV], and COVESA [COVESA]) towards autonomous vehicles in Intelligent Transportation Systems (ITS). An SDV is constructed by a software platform having a cloud-native system (e.g., Kubernetes [Kubernetes]) and has its internal network (e.g., a giga-bit Ethernet). For facilitating the easy and efficient configuration of networks, security, and applications in the SDV'S in-vehicle networks, an intent-based management is required. An intent-based management framework for SDVs is proposed by [I-D.jeong-opsawg-intent-based-sdv-framework]. This framework lets SDVs be configured and monitored by a vehicular cloud in terms of networks, security, and applications in SDVs. In this framework, SDVs can communicate with other SDVs and infrastructure nodes for safe driving and infotainment services in ITS.

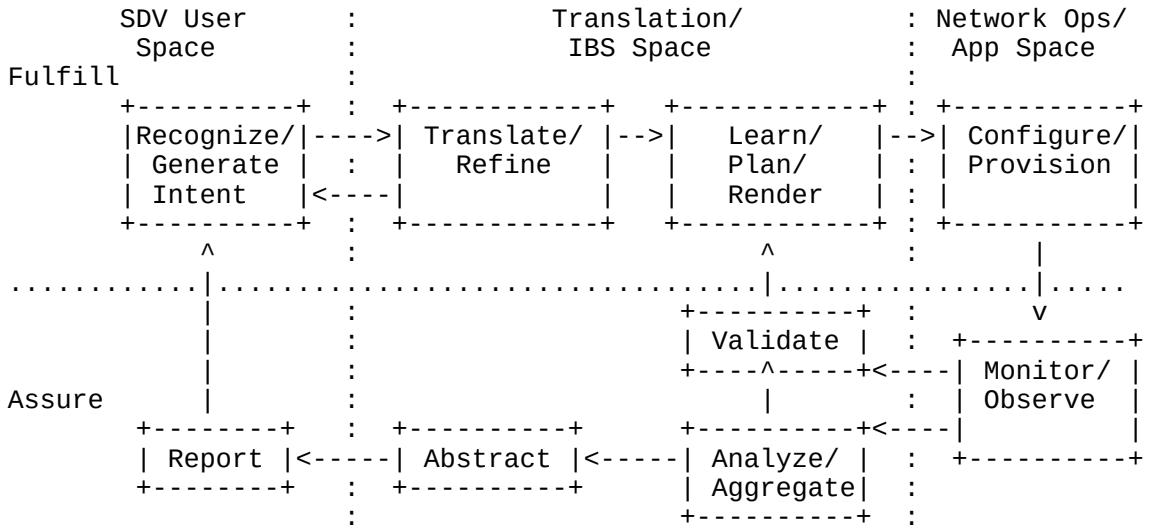

Figure 7: The Intent Life Cycle of IBS for SDV Management

According to the intent life cycle of an Intent-Based System (IBS) in [RFC9315], as shown in Figure 7, the intent life cycle of the IBS for SDVs can be enforced for SDV management. The life cycle consists of three spaces, namely SDV User Space, Translation & IBS Space, and Network Operations (Ops) & Application (App) Space. These spaces are divided into two phases in the life cycle space, such as fulfillment and assurance. The fulfillment phase (denoted as "Fulfill") pipelines the steps for an intent enforcement, such as intent input, translation/refinement, learning/planning/rendering, and

configuration/provisioning toward the target Service Functions (SFs), such as Network Functions (NFs) and Application Functions (AFs) in SDVs. On the other hand, the assurance phase (denoted as “Assure”) performs the steps for an intent validation and optimization by collecting final results of the intent fulfillment from the NFs and AFs for SDVs. If an action for the found problem is needed, the life cycle inserts a reconfigured policy or feedback information into the fulfillment phase or report a required action to an SDV User.

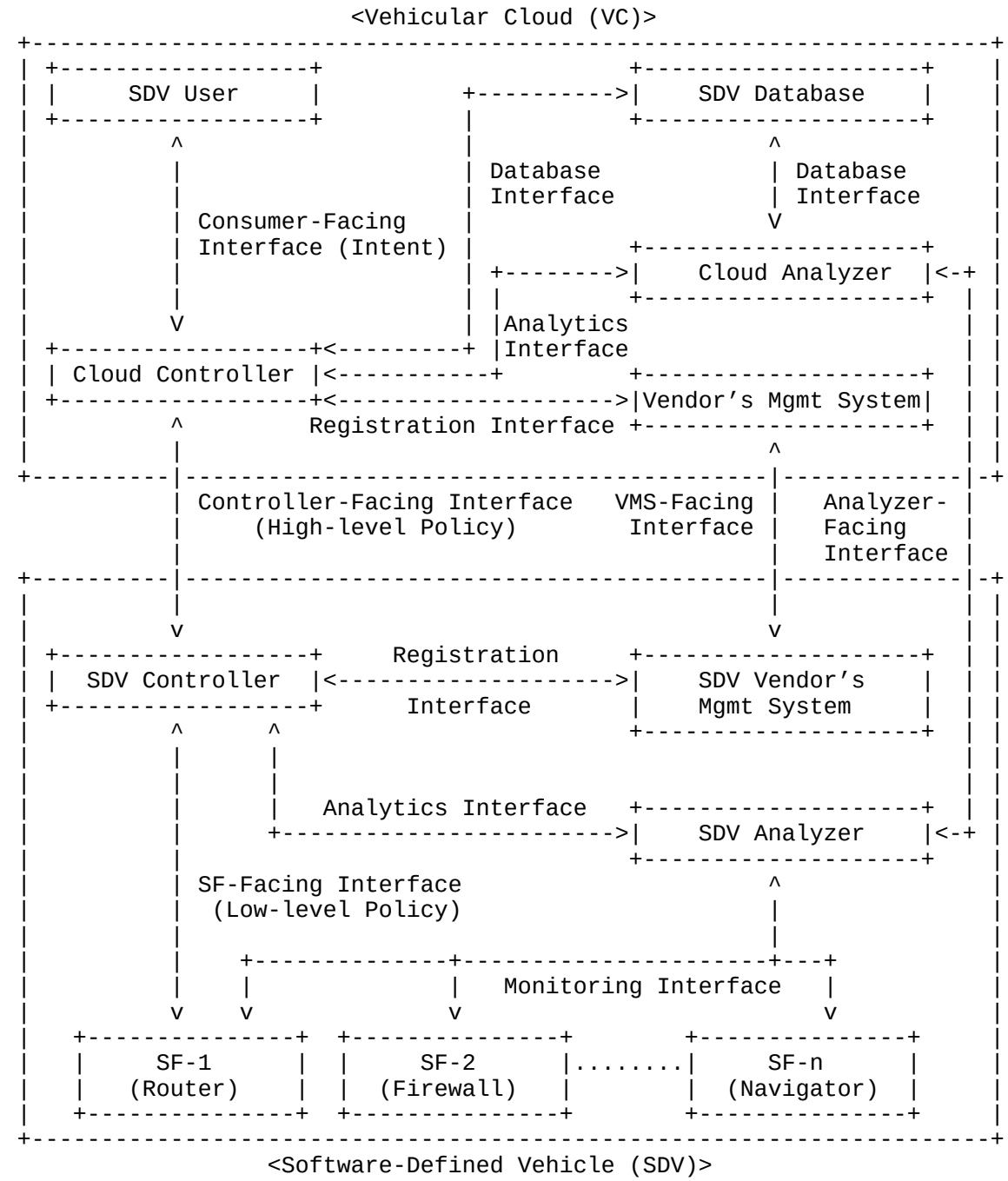
<Vehicular Cloud (VC)>
SDV User
SDV Database
Database Interface
Database Interface
Consumer-Facing Interface (Intent)
Cloud Analyzer
Analytics Interface
Cloud Controller
Vendor's Mgmt System
Registration Interface
Controller-Facing Interface (High-level Policy)
VMS-Facing Interface
Analyzer-Facing Interface
SDV Controller
Registration Interface
SDV Vendor's Mgmt System
Analytics Interface
SDV Analyzer
SF-Facing Interface (Low-level Policy)
Monitoring Interface
SF-1 (Router)
SF-2 (Firewall)
SF-n (Navigator)
<Software-Defined Vehicle (SDV)>

Figure 8: Intent-Based Management Framework for Software-Defined Vehicles

Figure 8 shows a framework of intent-based management for SDVs. The framework consists of a vehicular cloud and SDVs. The two parts of Vehicular Cloud and SDV borrow the components and interfaces of the I2NSF framework in [RFC8329][I-D.jeong-i2nsf-security-management-automation] and customize their components and interfaces for IBN-based SDV management.

For simplicy, Vehicular Cloud can be treated as SDV User (i.e., network administrator) like I2NSF User in [RFC8329]. In this case, the SDV framework in Figure 8 is similar to the I2NSF framework in [RFC8329].

3.6. IBN for Interconnection

New network capabilities based on programmability and virtualization are producing service situations where a connectivity-only approach is not sufficient. The increasing availability of computing capabilities internal to the networks, or attached to them, enable new scenarios where those capabilities can be consumed through the advertisement or exposure of these execution environments (i.e., compute, storage, and associated networking resources). In addition to that, even services or network functions could be advertised in order to make them available for interconnection.

Figure 9 captures the intent procedure for the fulfillment phase of the Interconnection Intent. Note that SLO, SLE, and SDP stand for “Service Level Objective”, “Service Level Expectation”, and “Service Demarcation Point”, respectively.

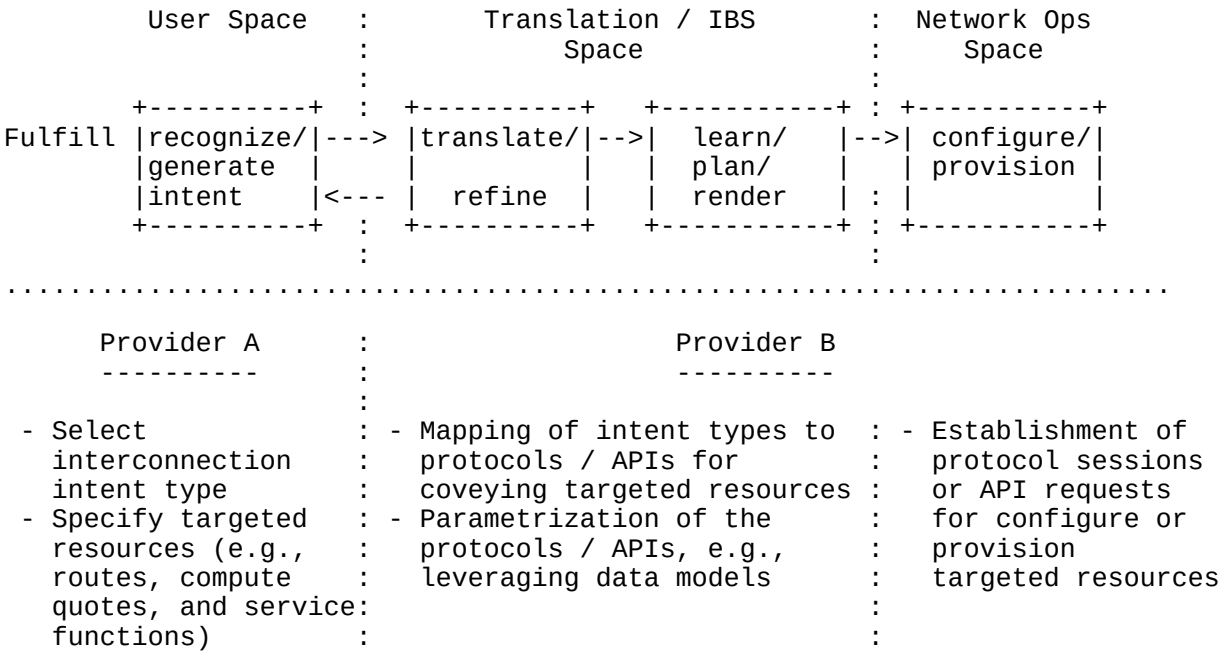

Figure 9: Fulfillment Phase of Interconnection Intent

Similarly, Figure 10 sketches the intent procedure for the assurance phase of the Interconnection Intent.

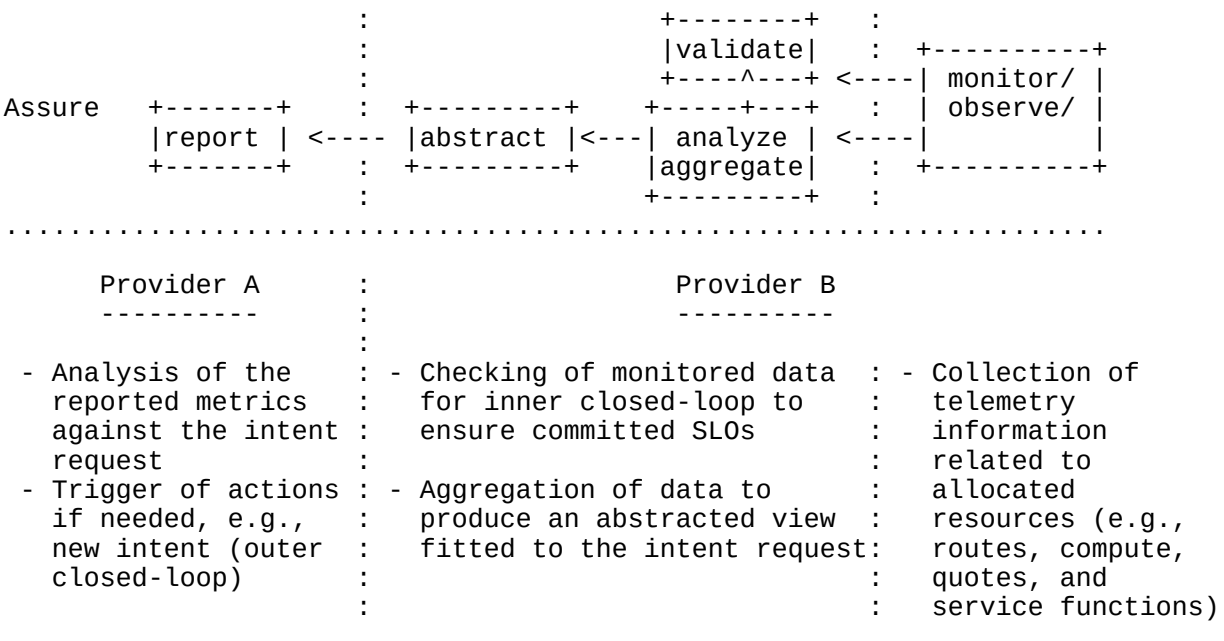

Figure 10: Assurance Phase of Interconnection Intent

In Figure 9 and Figure 10, both Fulfillment and Assurance phases are integral parts of the Interconnection Intent, respectively, according to the intent life cycle in [RFC9315]. For the more detailed discussion on an intent-based interconnection framework, refer to [I-D.contreras-nmrg-interconnection-intents].

3.7. IBN for IETF Network Slices

Network slicing is emerging as a future model for service offering in telecom operator networks. Conceptually, network slicing provides a customer with an apparent dedicated network which is built on top of logical (i.e., virtual) and/or physical functions and resources supported by a shared infrastructure. This infrastructure is provided by one or more telecom operators. As part of an End-to-End (E2E) network slice, it is expected to have a number of network slices at a transport level (referred as IETF network slices) providing the necessary connectivity to the rest of components of the E2E slice, e.g., mobile packet core slice.

With this respect, the GSMA [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). 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).

The previous slice templates provide a number of parameters that functionally characterizes the behavior of the network slice as expected by the slice customer. However, apart from the slice characteristics, further information is needed in order to request the realization of a slice towards the IETF Network Slice Controller (NSC), such as the identification of the slice endpoints and information about the virtual network topology expected to form the requested IETF Network Slice.

Figure 11 captures the intent procedure for the fulfillment phase of the IETF Network Slice Service Intent. Note that NBI and SBI stand for "Northbound Interface" and "Southbound Interface", respectively.

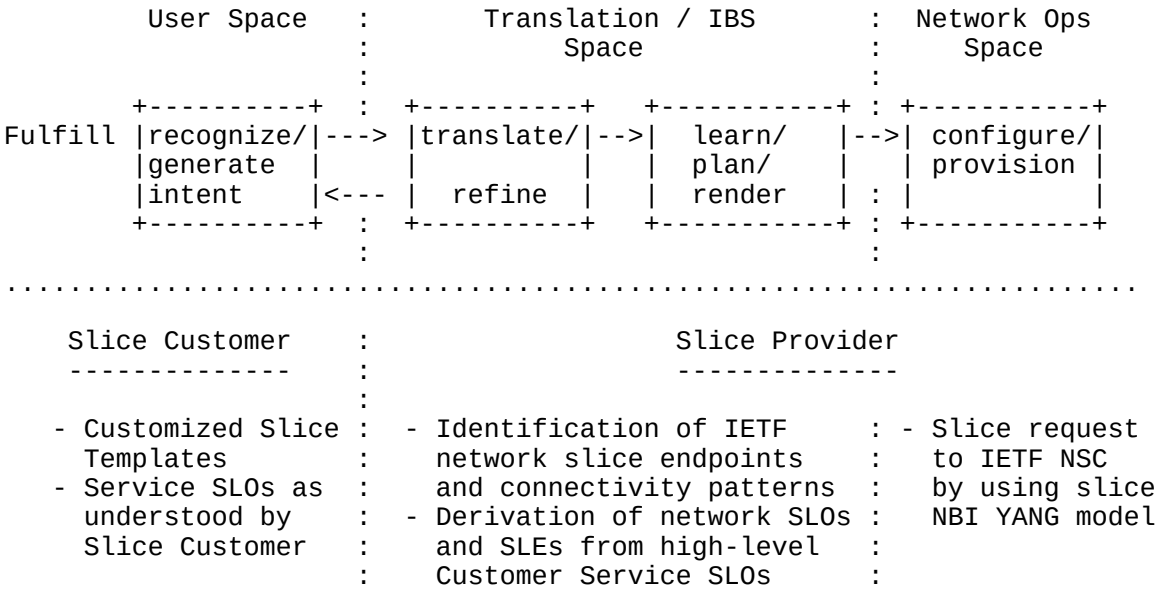

Figure 11: Fulfillment Phase of IETF Network Slice Service Intent

Similarly, Figure 12 sketches the intent procedure for the assurance phase of the IETF Network Slice Service Intent.

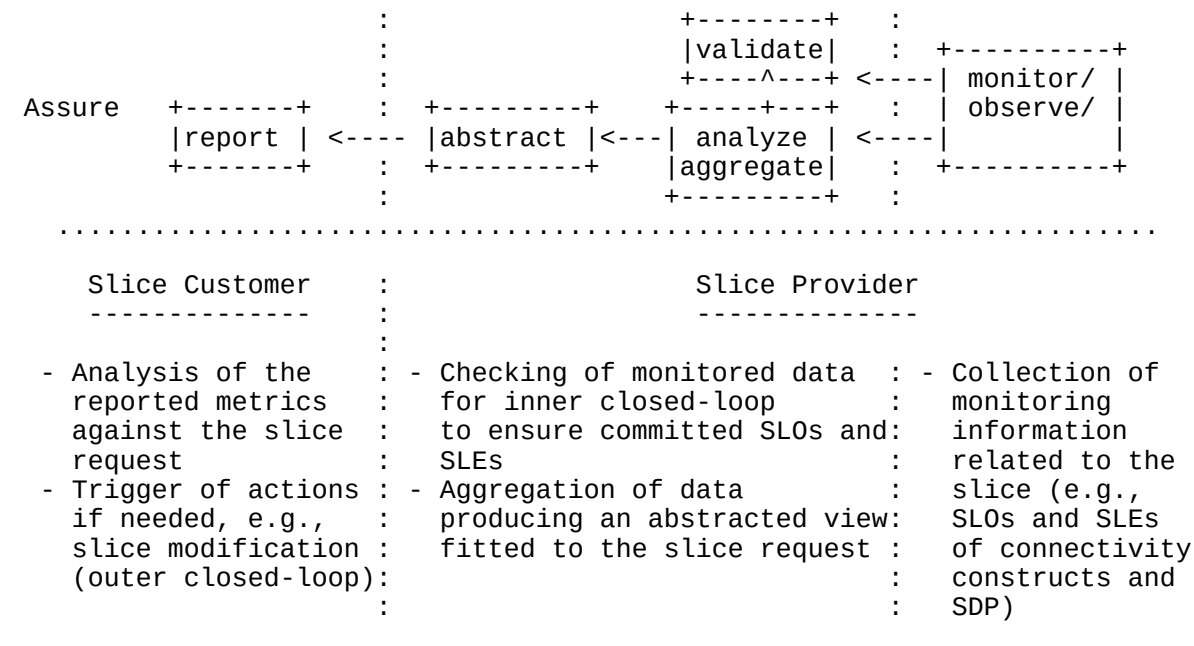

Figure 12: Assurance Phase of IETF Network Slice Service Intent

In Figure 11 and Figure 12, both Fulfillment and Assurance phases are integral parts of the IETF Network Slice Service Intent, respectively, according to the intent life cycle in [RFC9315]. For the more detailed discussion on an intent-based network slice service framework along with those terms, refer to [I-D.contreras-nmrg-transport-slice-intent].

3.8. IBN for Green Service Management

With the increasing need for sustainability in network services, Intent-Based Networking can be applied to enable customers to express green service objectives as network intents [I-D.contreras-nmrg-green-intent]. These intents allow customers to specify constraints and preferences related to energy consumption, carbon emissions, and the use of renewable energy in the provisioning and management of network services.

The green service intent includes attributes such as:

* Energy Consumption: Specifies maximum thresholds for total energy used by the network service.

* Energy Efficiency: Specifies minimum thresholds for energy efficiency metrics (e.g., bits per Joule).

* Carbon Emissions: Specifies limits on carbon intensity (grams CO2 per kWh) associated with the service.

* Use of Renewable Energy: Specifies minimum ratios of renewable energy sources powering the network functions.

These attributes can be specified individually or combined in a green intent, allowing flexible expression of sustainability goals.

Figure 13 captures the intent procedure for the fulfillment phase of the Green Intents.

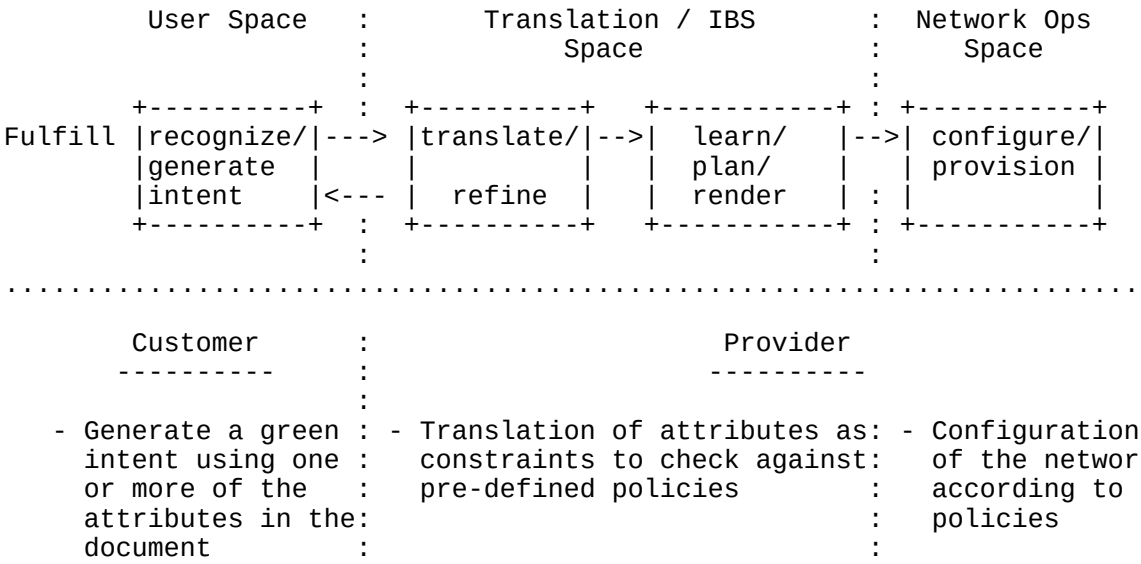

Figure 13: Fulfillment Phase of the Green Service Intent

The process begins when the customer generates a green intent that specifies the desired sustainability and energy-efficiency objectives for the network service. This intent is then interpreted by the intent translation module, which converts the high-level green objectives into concrete network policies and constraints. Next, the policy mapping module enforces these constraints by selecting appropriate network resources and configurations that align with the green goals, for instance, routing traffic through energy-efficient paths, leveraging equipment with lower carbon footprints, or prioritizing data centers powered by renewable energy. Finally, the configuration and provisioning modules deploy these configurations across the network infrastructure to realize the intended green service objectives.

Similarly, Figure 14 sketches the intent procedure for the assurance phase of the Green Service Intent.

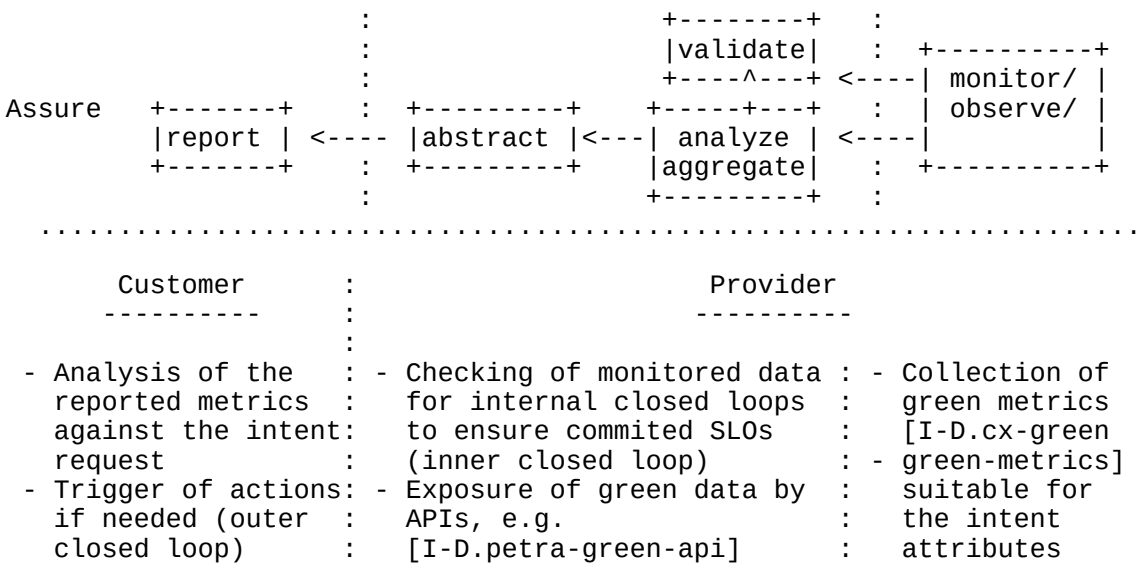

Figure 14: Assurance Phase of the Green Service Intent

In the case of assurance, monitoring modules continuously collect green metrics from various network elements. The gathered information is then analyzed by an analytics or validation module, which evaluates the network's performance and checks compliance with the established green intent objectives based on standardized green networking metrics. If any discrepancies or deviations from the intended goals are detected, the optimization module intervenes by adjusting network policies or reallocating resources to better align with the green objectives. In some cases, this process may also trigger a renegotiation or feedback loop with the customer to refine the intent or update the service parameters accordingly.

The green intents build upon existing green networking metrics and standardized APIs for energy and carbon reporting, such as those defined in [I-D.cx-green-green-metrics] and [I-D.petra-green-api]. Within this framework, the system must effectively manage trade-offs between traditional Quality of Service (QoS) objectives and green goals, maintaining service reliability and performance while optimizing energy efficiency and reduced emissions. Furthermore, the intent framework should incorporate negotiation and validation mechanisms, recognizing that network providers may only be able to partially fulfill green intents depending on their current infrastructure capabilities and operational policies.

4. Practice Learnings

4.1. Difficulties and Challenges

Some key learnings and takeaways can be extracted from the practices and implementation of IBN systems in different use cases. Commonly, there involve the following technical challenges in building IBN systems, incluing handling the dynamic and time variant nature of the network, the efficient management of cross-domain resources, and the reliability of automatic configuration, etc.

Let us take Service Function Chaining (called SFC) as an example to show three challenges as follows.

1. Stability in Dynamic Network Environments: For instance, in the space-terrestrial networks where the network topology is with frequent changes, it is essential to design efficient service function chain reconstruction and service recovery mechanisms. But how to guarantee the effectiveness of the chaining rule in these scenarios is still a challenge.

2. Collaborative Management of Cross-Domain SFC: To ensure the network intents across multi-domain networks, intent-based networks should be designed with a cross-domain orchestration and management framework to ensure an E2E optimization of Quality of Service (called QoS).

3. Deployment under Resource-Constrained Conditions: It is important to consider how to effectively deploy and manage these service function chains within limited resources. Methods such as intent negotiation can be introduced to optimize resource allocation in the SFC.

4.2. Future Research Directions

Although there have been extensive research achievements from academic, industrial, and standardization fields, there are the following three future research considerations.

1. Generic Intent model for Full Life-Cycle Assurance: It is necessary to construct an intent model for the full life-cycle from both top-to-down and down-to-top perspectives, including the intent input state, the intent execution state, and the intent completion state, etc, merged in a generic logic model. It makes sense of ensuring the E2E guaranteed implementation of any network intent and verifying the intent state through consistent mathematical logic.

2. Autonomous End-to-End Network Policy Generation: Intent-based networks should provide the network configuration policies to always well understand the requested network services in time, in particular towards various dynamic on-demand service requirements. Therefore, intent-based networks should make the network QoS satisfy the users' Quality of Experience (QoE) from a vertical perspective of a network protocol or different intent holders. Meanwhile, the current network is based on domain-specific local policy optimization, and it is hard to ensure an E2E QoS guarantee, in particular a cross-domain global optimization. Therefore, intent-based networks should provide an E2E optimization policies across multi-domain networking applications.

3. Intent Implementation with Large language Models (LLMs): Large language models (LLMs) such as Flan-T5 [Flan-T5] and GPT-3 [GPT-3] will play an important role in enhancing the accuracy of intent refinement, resulting from the powerful understanding capabilities of LLMs and the entity relationships in knowledge graphs [Knowledge-Graph]. It is also beneficial to network policy generation according to the network status. Although we have involved different kinds of AI models at each intent-based networks' stages, there still lack of generality and accuracy. Meanwhile, human interference is still in the full life-cycle of intent-based networks, and in the future the knowledge graph-assisted LLMs can further reduce the human intervention, and even make the human completely be out of the full life-cycle of the intent-based networks.

5. Discussion

This section discusses three aspects for the deployment of IBN systems on the real world. They are Multi-Domain Deployment, Network Digital Twin, and IBN with AI.

5.1. Multi-Domain Dichotomy for IBN

IBN shows two different aspects and relations with multi-domain concepts such as multi-domain intents and multi-domain intent resolution.

5.1.1. Multi-Domain Intents

Some network intents involve multiple domains. They can be explicit, especially when being expressed in a natural language, or implicit. The resolution of the former is generally straightforward. Probably a mapping is required to let the intent resolution system, e.g., one following the specification in [I-D.pedro-ite], to know the real identity of the domain mentioned in the intent.

On the other hand, the resolution of the implicit domains in network intents requires a much larger and consistent structure and mapping functions. They must be able to determine the involvement of multiple domains, and the reason must be clearly stated in the structures. For instance, if the network intent is resolved into a network service that involves a security function and the security function is only available at a different domain to the domain that is resolving the intent, the involvement of multiple domains is justified. Similarly, other scenarios must provide justifications for involving multiple domains implicitly.

5.1.2. Multi-Domain Intent Resolution

Regardless of a network intent being single-domain or multi-domain in Section 5.1.1, a network intent can be resolved by a standalone system, i.e., doing single-domain intent resolution, or by multiple interconnected systems, i.e., doing multi-domain intent resolution.

Involving multiple domains in the resolution of an intent has many benefits, such as using bigger knowledge bases and bigger network function structures. This is particularly beneficial for multi-domain intents. However, it also means that network management systems must consider additional security concerns and general domain information borders and policies for its transmission. This is being actively researched, but results are still early to say that a consistent multi-domain system can be built for network intent resolution.

5.2. The Integration of IBN and Network Digital Twin

As described in [I-D.irtf-nmrg-network-digital-twin-arch], the Network Digital Twin (NDT) can be an important enabler platform for implementing IBN systems and fostering their deployment. A user gives his (or her) intent to the network system with NDT. Through the closed-loop control with monitoring, validation, and adjustment in NDT, the IBN-based network management will be effectively performed with a minimum trial and error in the real networks. For more details on IBN interaction with Network Digital Twin, refer to Section 10 of [I-D.irtf-nmrg-network-digital-twin-arch].

5.3. IBN with AI

This section proposes some discussions on IBN with AI. AI techniques have been integrated by IBN, but there is still much space in researching on topics related to IBN with AI, such as different learning patterns, AI agents, and agentic AI.

5.3.1. Transfer Learning

[I-D.cgfabk-nmrg-ibn-generative-ai] describes how transfer learning techniques can be adopted to design generative AI specialized models for IBN.

IBN represents a paradigm shift in network management, aiming to bridge the gap between business objectives and network configurations. IBN allows operators to specify high-level intents, which the system then automatically translates, enforces, and continuously validates. Generative AI, particularly Large Language Models (LLMs), can enhance IBN by automating intent parsing, policy generation, and network troubleshooting. LLMs can understand natural language intents, generate high-level policies, and adapt the policies in real time. Transfer learning enables pretrained models to adapt to specific tasks with significantly less data and computational resources. In the context of IBN, this approach offers a dual advantage: (i) enhancing the efficiency of model training and (ii) improving the reliability of intent recognition and execution.

5.3.2. AI Agent-Enabled IBN

In the future, IBN will be closely intertwined with AI Agents and Multi-Agent systems. Multi-Agent systems, equipped with capabilities of distributed perception, collaborative decision-making, and autonomous execution, will serve as the core technical engine for IBN to achieve full-process automation. For example, the Confucius framework in [Confucius] has proven that multi-agent collaboration can improve the accuracy of network management tasks by over 34%.

The core research issues in the integration of the IBN with AI focus on three dimensions as follows:

1. Accurate translation and decomposition of intents: This dimension considers the accuracy of intent translation. It needs to resolve the ambiguity of intents expressed in a natural language.

2. Collaboration and management of multi-agents: This dimension considers network scalability. As the network scale expands, the increase in the number of agents leads to issues such as decision consistency and resource competition. Additionally, it is necessary to handle the reasonable decomposition and dependency management of complex tasks among multiple agents.

3. System reliability and security: This dimension considers the stability and cybersecurity. This not only includes the logical verification of instructions generated by AI Agents (e.g., Domain-Specific Language (DSL) syntax compliance) but also involves issues such as data privacy protection, malicious behavior identification, and Byzantine fault tolerance in agent interactions. Note that the Byzantine fault tolerance allows the IBN systems to keep operating correctly or reach right consensus even if some components of the IBN systmes are malicious or faulty.

Potential research directions and technologies can be as follows:

* Enhanced intent understanding: It optimizes intent parsing by combining domain knowledge graphs [Knowledge-Graph] and Retrieval-Augmented Generation (RAG) [RAG]. It can realize simulation verification and preview of intents (e.g., What-If analysis) through digital twins.

* Efficient multi-agent collaborative architectures: It adopts a hierarchical agent design to reduce cross-layer communication overhead. Federated learning may enable dynamic task scheduling and parallel execution while protecting local data in each agent.

* Trusted agent technologies: It includes a multi-layer verification mechanism of “agent pre-verification” and “manual approval” and also abnormal behavior detection algorithms based on traffic fingerprints.

* Performance acceleration and resource optimization technologies: It matches the computing power needs of agents with network loads through dynamic resource scheduling algorithms to improve the operational efficiency of the IBN systems.

6. Security Considerations

There are many considerations on security. First, the IBN systems should be strong and resilient against variable security attacks from outsider attacks (e.g., Distributed-Denial-of-Service (DDoS) attacks and virus) and to insider attacks (e.g., supply chain attacks).

A malicious intent can break down the whole IBN system if the analysis of each intent for the impact on the IBN system is not performed appropriately on time. Thus, the IBN defense system should execute both the security check for an intent during the Fulfillment Phase and the attack monitoring during the Assurance Phase. Such security check and attack monitoring need to be performed by the collaboration between AI agents and network administrators.

7. IANA Considerations

This document has no requests to IANA.

8. References

8.1. 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>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

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

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

[RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <https://www.rfc-editor.org/info/rfc8329>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/info/rfc9315>.

[RFC9316] Li, C., Havel, O., Olariu, A., Martinez-Julia, P., Nobre, J., and D. Lopez, "Intent Classification", RFC 9316, DOI 10.17487/RFC9316, October 2022, <https://www.rfc-editor.org/info/rfc9316>.

[RFC9326] Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", RFC 9326, DOI 10.17487/RFC9326, November 2022, <https://www.rfc-editor.org/info/rfc9326>.

[RFC9341] Fioccola, G., Ed., Cociglio, M., Mirsky, G., Mizrahi, T., and T. Zhou, "Alternate-Marking Method", RFC 9341, DOI 10.17487/RFC9341, December 2022, <https://www.rfc-editor.org/info/rfc9341>.

[RFC9342] Fioccola, G., Ed., Cociglio, M., Sapio, A., Sisto, R., and T. Zhou, "Clustered Alternate-Marking Method", RFC 9342, DOI 10.17487/RFC9342, December 2022, <https://www.rfc-editor.org/info/rfc9342>.

8.2. Informative References

[AI-Intent-Network] Njah, Y., Leivadeas, A., and M. Falkner, "An AI-Driven Intent-Based Network Architecture", IEEE Communications Magazine, Volume 63, Issue 4, April 2025, <https://doi.org/10.1109/MCOM.001.2400143>.

[AUTOSAR] AUTOSAR, "Adaptive Platform", <https://www.autosar.org/standards/adaptive-platform>.

[Circular-Reasoning]
Rips, L., "Circular Reasoning", Wiley Cognitive Science, Volume 26, Issue 6, November 2002, <https://doi.org/10.1016/S0364-0213(02)00085-X>.

[Confucius]
Wang, Z., "Intent-Driven Network Management with Multi-Agent LLMs: The Confucius Framework", 2025, <https://doi.org/10.1145/3718958.3750537>.

[COVESA] COVESA, "Connected Vehicle Systems Alliance", <https://covesa.global/>.

[DRL] Luong, N., Hoang, D., Gong, S., Niyato, D., Wang, P., and Y. Liang, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey", IEEE Communications Surveys & Tutorials, Volume 21, Issue 4, October 2019, <https://ieeexplore.ieee.org/document/8714026>.

[Eclipse-SDV]
Eclipse, "Eclipse Software Defined Vehicle Working Group Charter", <https://www.eclipse.org/org/workinggroups/sdv-charter.php>.

[ETSI-NFV] ETSI, "Network Functions Virtualisation (NFV); Architectural Framework", ETSI GS NFV 002 V1.2.1, December 2014, <https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf>.

[ETSI-NFV-Release-2]
ETSI, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Architectural Framework Specification", ETSI GS NFV 006 V2.1.1, January 2021, <https://www.etsi.org/deliver/etsi_gs/nfv/001_099/006/02.01.01_60/gs_nfv006v020101p.pdf>.

[Flan-T5] Chung, H., "Scaling Instruction-Finetuned Language Models", arXiv arXiv:2210.11416, October 2022, <https://arxiv.org/abs/2210.11416>.

[GPT-3] Brown, T., "Language Models are Few-Shot Learners", arXiv arXiv:2005.14165, May 2020, <https://arxiv.org/abs/2005.14165>.

[GSMA] GSMA, "Global System for Mobile Communications Association", <https://www.gsma.com>.

[I-D.cgfabk-nmrg-ibn-generative-ai]
Cassara', P., Gotta, Fioccola, G., Artigiani, A., Burrai, R., Kolaj, E., Martalo', M., and V. Pilloni, "Generative AI for Intent-Based Networking", Work in Progress, Internet-Draft, draft-cgfabk-nmrg-ibn-generative-ai-01, 17 October 2025, <https://datatracker.ietf.org/doc/html/draft-cgfabk-nmrg-ibn-generative-ai-01>.

[I-D.contreras-nmrg-green-intent]
Contreras, L. M. and G. S. Illan, "Intent for Green Services", Work in Progress, Internet-Draft, draft-contreras-nmrg-green-intent-01, 17 March 2025, <https://datatracker.ietf.org/doc/html/draft-contreras-nmrg-green-intent-01>.

[I-D.contreras-nmrg-interconnection-intents]
Contreras, L. M., Lucente, P., and T. H. Velivassaki, "Interconnection Intents", Work in Progress, Internet-Draft, draft-contreras-nmrg-interconnection-intents-05, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-contreras-nmrg-interconnection-intents-05>.

[I-D.contreras-nmrg-transport-slice-intent]
Contreras, L. M., Demestichas, P., and J. Tantsura, "IETF Network Slice Intent", Work in Progress, Internet-Draft, draft-contreras-nmrg-transport-slice-intent-07, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-contreras-nmrg-transport-slice-intent-07>.

[I-D.cx-green-green-metrics]
Clemm, A., Pignataro, C., Schooler, E., Ciavaglia, L., Rezaki, A., Mirsky, G., and J. Tantsura, "Green Networking Metrics for Environmentally Sustainable Networking", Work in Progress, Internet-Draft, draft-cx-green-green-metrics-00, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-cx-green-green-metrics-00>.

[I-D.fz-ippm-on-path-telemetry-yang]
Fioccola, G. and T. Zhou, "On-path Telemetry YANG Data Model", Work in Progress, Internet-Draft, draft-fz-ippm-on-path-telemetry-yang-01, 20 December 2024, <https://datatracker.ietf.org/doc/html/draft-fz-ippm-on-path-telemetry-yang-01>.

[I-D.gu-nmrg-intent-translator]
Gu, M. and J. P. Jeong, "An Intent Translation Framework for Internet of Things", Work in Progress, Internet-Draft,

draft-gu-nmrg-intent-translator-02, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-gu-nmrg-intent-translator-02>.

[I-D.ietf-opsawg-ipfix-alt-mark]
Graf, T., Fioccola, G., Zhou, T., Zhu, Y., and M. Cociglio, "IP Flow Information Export (IPFIX) Alternate-Marking Information Elements", Work in Progress, Internet-Draft, draft-ietf-opsawg-ipfix-alt-mark-04, 16 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-ipfix-alt-mark-04>.

[I-D.ietf-spring-sr-policy-yang]
Saleh, T., Raza, S. K., Zhuang, S., Matsushima, S., and V. P. Beeram, "YANG Data Model for Segment Routing Policy", Work in Progress, Internet-Draft, draft-ietf-spring-sr-policy-yang-06, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-policy-yang-06>.

[I-D.irtf-nmrg-network-digital-twin-arch]
Zhou, C., Yang, H., Duan, X., Lopez, D., Pastor, A., Wu, Q., Boucadair, M., and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-11, 6 July 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-11>.

[I-D.jeong-i2nsf-security-management-automation]
Jeong, J. P., Lingga, P., Jung-Soo, J., Lopez, D., and S. Hares, "An I2NSF Framework for Security Management Automation in Cloud-Based Security Systems", Work in Progress, Internet-Draft, draft-jeong-i2nsf-security-management-automation-08, 26 July 2024, <https://datatracker.ietf.org/doc/html/draft-jeong-i2nsf-security-management-automation-08>.

[I-D.jeong-nmrg-ibn-network-management-automation]
Jeong, J. P., Ahn, Y., Gu, M., Kim, Y., and J. Jung-Soo, "Intent-Based Network Management Automation in 5G Networks", Work in Progress, Internet-Draft, draft-jeong-nmrg-ibn-network-management-automation-06, 9 June 2025, <https://datatracker.ietf.org/doc/html/draft-jeong-nmrg-ibn-network-management-automation-06>.

[I-D.jeong-opsawg-intent-based-sdv-framework]
Jeong, J. P., Shen, Y. C., Ahn, Y., and M. Gu, "An Intent-Based Management Framework for Software-Defined Vehicles in Intelligent Transportation Systems", Work in Progress, Internet-Draft, draft-jeong-opsawg-intent-based-sdv-framework-04, 9 June 2025, <https://datatracker.ietf.org/doc/html/draft-jeong-opsawg-intent-based-sdv-framework-04>.

[I-D.park-nmrg-ibn-network-management-srv6]
Jung-Soo, J., Choi, Y., and J. P. Jeong, "Intent-Based Network Management in SRv6 Networks", Work in Progress, Internet-Draft, draft-park-nmrg-ibn-network-management-srv6-02, 24 June 2024, <https://datatracker.ietf.org/doc/html/draft-park-nmrg-ibn-network-management-srv6-02>.

[I-D.pedro-ite]
Martinez-Julia, P., Jeong, J. P., Miyasaka, T., and D. Lopez, "Intent Translation Engine for Intent-Based Networking", Work in Progress, Internet-Draft, draft-pedro-ite-02, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-pedro-ite-02>.

[I-D.petra-green-api]
Rodriguez-Natal, A., Contreras, L. M., Palmero, M. P., Lindblad, J., and A. G. Snchez, "Path Energy Traffic Ratio API (PETRA)", Work in Progress, Internet-Draft, draft-petra-green-api-02, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-petra-green-api-02>.

[I-D.xia-sdnrg-nemo-language]
Xia, Y., Jiang, S., Zhou, T., Hares, S., and Y. Zhang, "NEMO (NEtwork MOdeling) Language", Work in Progress, Internet-Draft, draft-xia-sdnrg-nemo-language-04, 14 April 2016, <https://datatracker.ietf.org/doc/html/draft-xia-sdnrg-nemo-language-04>.

[I-D.ydt-ippm-alt-mark-yang]
Graf, T., Wang, M., Fioccola, G., Zhou, T., and X. Min, "A YANG Data Model for the Alternate Marking Method", Work in Progress, Internet-Draft, draft-ydt-ippm-alt-mark-yang-03, 2 September 2024, <https://datatracker.ietf.org/doc/html/draft-ydt-ippm-alt-mark-yang-03>.

[INT] Tan, L., Su, W., Zhang, W., Lv, J., Zhang, Z., Miao, J., Liu, X., and N. Li, "In-band Network Telemetry: A Survey", Elsevier Computer Networks, Volume 186, February 2021, <https://doi.org/10.1016/j.comnet.2020.107763>.

[IntFlow] Shi, Q., Wang, F., and D. Feng, "IntFlow: Integrating Per-Packet and Per-Flowlet Switching Strategy for Load Balancing in Datacenter Networks", IEEE Transactions on Network and Service Management, Volume 17, Issue 3, September 2020, <https://ieeexplore.ieee.org/document/9079931>.

[Knowledge-Graph] Ji, S., Pan, S., Cambria, E., Marttinen, P., and P. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications", IEEE Transactions on Neural Networks and Learning Systems, Volume 33, Issue 2, February 2022, <https://ieeexplore.ieee.org/document/9416312>.

[Kubernetes] Kubernetes, "K8s: Cloud Native Computing Platform", <https://kubernetes.io/>.

[Nile] Jacobs, A., Pfitcher, R., Ferreira, R., and L. Granville, "Refining Network Intents for Self-Driving Networks", ACM SIGCOMM 2018 Workshop on Self-Driving Networks (SelfDN), August 2018, <https://doi.org/10.1145/3229584.3229590>.

[OpenStack] OpenStack, "OpenStack: Open Source Cloud Computing Infrastructure", <https://www.openstack.org>.

[RAG] Lewis, P., Ed., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", The 34th International Conference on Neural Information Processing Systems (NIPS 2020), December 2020, <https://dl.acm.org/doi/abs/10.5555/3495724.3496517>.

[REST] Fielding, R. and R. Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology, Volume 2, Issue 2, May 2002, <https://dl.acm.org/doi/10.1145/514183.514185>.

[SPT] Lingga, P., Jeong, J., Yang, J., and J. Kim, "SPT: Security Policy Translator for Network Security Functions in Cloud-Based Security Services", IEEE Transactions on Dependable and Secure Computing, Volume 21, Issue 6, November 2024, <https://ieeexplore.ieee.org/document/9416312>.

[TS-28-312] ETSI, "Intent Driven Management Services for Mobile Networks", ETSI TS 28.312 V18.5.0, September 2024, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3554>.

[YAML] YAML Language Development Team, "Yet Another Markup Language (YAML) version 1.2", October 2021, <https://yaml.org/spec/1.2.2/>.

Acknowledgments

This work of Jaehoon Paul Jeong is supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. RS-2024-00398199 and No. RS-2022-II221015).

The work of Luis M. Contreras has been partially funded by the European Union under Horizon Europe projects NEMO (NExt generation Meta Operating system) grant number 101070118, and SNS 6Green (Green Technologies For 5/6G Service-Based Architectures) grant agreement 101096925.

Contributors

The following people have substantially contributed to this document as co-authors:

Hongwei Yang
China Mobile
Email: yanghongwei@chinamobile.com

Yiwen Shen
Ajou University
Email: chrisshen@ajou.ac.kr

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

Yoseop Ahn
Sungkyunkwan University
Email: ahnjs124@skku.edu

Mose Gu
Sungkyunkwan University
Email: rna0415@skku.edu

Younghan Kim
Soongsil University
Email: younghak@ssu.ac.kr

Jung-Soo Park
Electronics and Telecommunications Research Institute
Email: pjs@etri.re.kr

Yun-Chul Choi
Electronics and Telecommunications Research Institute
Email: cyc79@etri.re.kr

Guillermo Sanchez Illan
Telefonica
Email: guillermo.sanchezillan@telefonica.com

Authors' Addresses

Kehan Yao (editor)
China Mobile
Beijing
100053
China
Email: yaokehan@chinamobile.com

Danyang Chen (editor)
China Mobile
Beijing
100053
China
Email: chendanyang@chinamobile.com

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Qin Wu
Huawei
Email: bill.wu@huawei.com

Chungang Yang
Xidian University
Email: cgyang@xidian.edu.cn

Luis M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

Giuseppe Fioccola
Huawei
Email: giuseppe.fioccola@huawei.com