Abstract

Intent was defined as an abstract high-level policy (or instruction) used to trigger network operations (not only configuration aspects, but also continuous tuning to adjust the measured/actual network state to an expected state). This document describes the intent-driven management architecture, its key elements, and interfaces.
The network must evolve from a static resource management system to a more dynamic system that consistently and continuously meets business goals.

The concept of Intent-driven management was proposed to deal with this issue. This is a networking model in which high level abstracted business requirements or business request, i.e., "intents", are formally defined and then the network continuously monitors whether these "intentions" are met. Such high-level abstracted business request are translated into actions (including configuration) and maintained by a set of management function blocks in one or more network management systems.

[RFC7575] defines Intent as an abstract high-level policy used to operate the network. Intent management system includes an interface for users/applications to input requests and an engine to translate the intents into the network actions and manage their lifecycle. This document describes an Intent-driven architecture, its elements, and interfaces.

2. Acronyms

CLI: Command Line Interface
SDO: Standards Development Organization
VPN: Virtual Private Network

3. Framework for Generic Intent-driven Management

This section briefly describes the design and operation of the Intent-driven management framework.

3.1. Overview

Figure 1 shows a simplified functional architecture of how Intent is used to manage a network (e.g., network element configuration failure, performance and security management, etc.).
In the architecture of intent-driven management, the intent orchestration layer (hereinafter referred to as intent layer) consists of four functional modules, which are "intent management", "translation", "verification", and "decision" module.
Combined with the request/experience from upper layer (including operators, administrators, users or applications. The different roles of identities in the network may cause different levels of abstraction involved in the intent request. The term "user" in this framework refers to all the entities who make intent requests), the intent-driven management system can collect the intent request, map out appropriate policy, verify correctness and then deploy configuration automatically.

When the configuration is executed in the network (i.e. network element) the verification module should verify the validity of configuration implementation based on real-time status. Therefore, the effectiveness of the configuration is repeatedly verified and monitored to ensure that the user's intent is effectively processed and implemented. In case of any unexpected situation is encountered, it can automatically make remedial measures instantly, and provide feedback to the user to achieve a complete lifecycle.

In the intent-driven networking, the application layer or service layer no longer directly interacts with network layer, but communicates indirectly with the network layer through the intermediate intent layer with certain technical agreement.

```
+---------------------------------+
    |          Intent model          |
    +-------------------------------+
          v
+-----------------+-----------------+
    |     Service / Policy model     |
    +-----------------+-----------------+
          v
+---------------+-----------------+
    |           Device model(s)      |
    +-------------------------------+
```

Figure 2 The Model Process of Intent Transformation

Figure 2 presents the overall intent transformation process. At the entrance, the user or administrator needs to express their desired "intent" according to the intent model, which is collected by management module in a high-level language.
With the help of policy database, the translation module and decision module can analyze the intent model and transform it into specific strategy or policy, namely service / policy model.

Eventually the policy will be mapped into concrete configuration action which can be deployed and executed in device level, therefore referred as device model.

The network elements used in this framework are:

Control Layer, which represents one or more entities that are able to control the operation and management of a network infrastructure (e.g., a network topology that consists of Network Elements).

Network Element, which can interact with local or remote Control Layer in order to exchange information, such as configuration information, policy enforcement capabilities, and network status.

3.2. Sample Operation

3.2.1 Intent layer

(1) Intent management

After receiving the user/application intent from the business layer, the intent orchestration layer needs to manage and coordinate these intents. Eligible intent will be further submitted to the intent translation module for analysis and processing, and the ineligible intent will be fed back to the user by the management module, for further modification and adjustment; the management module also needs to interact with the decision module, and may receive the negative feedback from decision module when the configuration execution does not work as expected, in which cases the management module would require translation module to do secondary processing (take optimization procedure or remedial solution).

In the process of intent demand, the module of the intent layer not only processes once, but continuously verifies and updates the configuration in the closed loop to finally realize the user's intent demand, so the management module also needs to regulate the life cycle of the intent demand.

In addition, because the intent needs to be from different identities, they contain different levels of demands and
abstractions. The abstraction level of the intent demand can be roughly divided into the following levels:

Device level (Level 0): In a traditional networking, the configuration of the device can be manually performed by the network administrator, such as configuring the ACL (Access Control List) of the device to allow an IP X to access a port of the server;

Network level (level 1): In some partially automated networking, network administrators usually do not manually configure specific commands on specific underlying devices. Instead, they set corresponding network permission settings through the controller, such as setting an IP X can access a port of the server through an access point AP.

Abstract intent level 1 (level 2): In a fully automated networking, we hope that the network configuration can be completed in the form of intent demands, but the intent demands should include specific details, for example: set "a user A can access a specific service of a server in the campus from the internal network or the external network."

Abstract intent level 2 (level 3): In this level, we hope that the network can be optimized spontaneously, and the intent defined at this level is more abstract than the previous level, for example, a specific group of users can access a specific service in any way;

Abstract intent level 3 (level 4): In this level, the network belongs to a part of the autonomous network, and the network can automatically decide a more appropriate configuration scheme, and the expression of the intention can be more abstract, for example, "Every user can access a service in a suitable way;"

Abstract intent level 4 (level 5): At the highest level of the intent abstraction model, we hope that the network can achieve full autonomy, automatically decision-making and spontaneous optimization based on real-time network state information anytime and anywhere. This level expresses the expectation that the network will decide the solution autonomously and never fail.

According to different levels of intent demands, the configuration should also be classified into different levels. Therefore, the management module needs to handle and treat the levels of "intent"-"configuration" one by one.

(2) Intent translation
The translation module needs to analyze the intent and formulate the corresponding configuration based on network status and analysis results in the intent-policy repository, and then output the configuration plan to the verification module. When the negative feedback of the management module is received subsequently, the intent analysis and translation module is also responsible for the dynamic adjustment and re-enactment of the strategy to finally achieve the user’s intention; if the positive feedback is received, the configuration is handed over to the control layer. The scope of intent includes several aspects including access control, bandwidth management, QoS levels, security issue, energy consumption control, etc., which enables application providers and users to reasonably mobilize resources and self-service as needed.

In the process of analyzing intent demands, first, the translation module needs to translate and split the semantics contained in the intent demand, because user-initiated intent demands are often complex and more natural-oriented, and to facilitate subsequent processing, the translation module needs to be translated and split according to a specific model. In this step, the method and result of natural language processing (NLP) in the field of artificial intelligence (AI) can be used to train the AI agent as an auxiliary, and the analysis result of the AI auxiliary module is used as a reference for the translation module.

(3) Verification module

The verification module has two functions: one is the execution verification of the configuration, that is, the configuration plan obtained during the translation process needs to be verified whether it can be executed in the actual network environment according to the real-time network status, and provide relevant information to the decision module as feedback; the second function is validity verification, for the reason that, when the configuration is actually executed, the network status may not change as expected, in which cases, it is necessary to verify whether the execution of the configuration works out as expected and whether the configuration meets the user’s intent. If the expected effect is achieved, the verification result is fed back to the decision module; if the verification fails, the "intent"-"configuration" translation procedure needs to be performed again.

The network is dynamically changing, so network verification should be continuous in real time. The verification module is responsible for monitoring the global information in the network, especially those that have an impact on the intent of the implementation. The
verification module needs to quickly reflect the monitoring information to the decision module to ensure that the execution of the configuration does not cause conflicts.

(4) Decision module

The decision module is responsible for the overall monitoring of the network status and configuration. It can process the data transmitted by the verification module, and then decide whether the configuration can be delivered to control layer. The decision module supports empirical knowledge from top administrators to help make decisions. When the user’s intent is not implemented or the network status is getting abnormal, the decision module needs to make optimization or remedial measures by notifying the intent management module and the translation module to make prompt response.

5 Policy repository

Regarding the storage of the intent model and the policy model by the policy repository, we must consider the different levels of the intent model and the policy model, and also consider under what conditions the data will be updated. Policy repository as a database of "intent" - "configuration" information should be able to interact with the intent management module and the translation module, provide relevant data for the intent model for the intent management module, and provide mapping between the "intent" and "configuration" for the translation module. According to the identity of the user, the abstraction level involved in different intent demands also needs to be graded. For the policy repository, the mapping between "intent" and "configuration" should be one-to-one correspondence at the abstract level.

When the translation module finds that an "intent"-"configuration" mapping relationship exists in the policy repository, it cannot be directly deployed to the network layer, but also through the executable verification of the verification module, and then the decision-making module makes a decision whether to issue the decision, because the execution of the configuration needs to consider the actual network environment and state. When the management module or the translation module obtains the intent model or the configuration model that is not in the policy repository, after the intent demand is implemented, the new mapping information should also be updated by the management module into the policy repository to ensure subsequent use.

3.2.2 Control layer
(1) Configuration delivery

In an intent-driven networking, in consideration of the configuration set of the intent layer output, the intent layer performs configuration control and further delivery to the control layer through the downward interface to ensure that the configuration can be smoothly delivered to the network layer for execution. The configuration delivery module requires further fine-grained distribution, according to different network structures such as wired access or wireless access, virtual network infrastructure or physical network facilities, to achieve the ability of different networks to work together.

(2) Network status collection

The downward interface of the control layer should be able to perceive the physical or logical topology of the underlying network layer, network traffic, network device status, etc., and the upward interface can transmit information to the intent layer to assist the intent orchestration layer to formulate a reasonable scheduling strategy and better utilize the capabilities of the intelligent communication network.

3.2.3 Network layer

(1) Configuration execution

The configuration execution function can be configured with pre-defined rules and templates, or can be fully dynamically configured according to the controller: for example, through the control protocol between the controller and the network forwarding device, the network layer receives the result of the configuration control function and performs the forwarding and processing of the configuration.

(2) Network status awareness

In an intent-driven intelligent communication networking, the sensing information collection function supports the collection of network topology information, network traffic information, and business path information, and can interact in real time through dynamic protocols. Especially when the configuration is actually executed, feedback information can be output to the network state collection function for subsequent verification.
4. Security Considerations

This document does not have any Security Considerations.

5. IANA Considerations

This document has no actions for IANA.

6. Contributors

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

Xueying Han
Institute of Network Technology,
Beijing University of Posts and Telecommunications,
Xitucheng Road, Haidian District, Beijing 100876

Email: hanxueying@bupt.edu.cn

Ruoshui Zhang
China Telecom,
No.118, Xizhimennei Street, Beijing 100035
P. R. China

Email: zrs_1995@bupt.edu.cn

Qiuhuan Ren
China Telecom,
No.118, Xizhimennei Street, Beijing 100035
P. R. China

Email: renqiuhuan123@bupt.edu.cn
7. Acknowledgments

This document has benefited from reviews, suggestions, comments and proposed text provided by the following members, listed in alphabetical order: Mohamed Boucadair.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Qiong Sun
China Telecom
No.118, Xizhimennei Street, Beijing 100035
P. R. China

Email: sunqiong.bri@chinatelecom.cn

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District, Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

Kun Xie
School of Software Engineering,
Beijing University of Posts and Telecommunications
Xitucheng Road, Haidian District, Beijing 100876
P.R. China

Email: pat@bupt.edu.cn