

Network Management Operations
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

X. Zhao
CAICT
M. Wang
China Mobile
B. Wu
Huawei
D. Ceccarelli
Cisco
J. Zhou
ZTE
H. Zheng
Huawei
7 July 2025

AI based Network Management Agent(NMA): Concepts and Architecture
draft-zhao-nmop-network-management-agent-02

Abstract

With the development of AI(Artificial Intelligence) technology, large model have shown significant advantages and great potential in recognition, understanding, decision-making, and generation, and can well match the self-intelligent network management requirements for the goal of autonomous network or Intent-based Networking, and can be used as one of the potential driving technologies to drive high-level autonomous networks. When introducing AI for network management, how to integrate AI technology and deal with the relationship with the existing network management entity (such as network controller) is the focus of research and standardization.

This document presents the concept of AI based network management agent(NMA), provides the basic definition and reference architecture of NMA, discusses the relationship of NMA with traditional network controller or other network management entity by exploring the deployment mode of NMA, and proposes the common processing flow and typical application scenarios of NMA.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Network Management Operations Working Group mailing list (nmop@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/nmop/>.

Source for this draft and an issue tracker can be found at <https://github.com/ietf-wg-nmop/draft-ietf-nmop-digital-map-concept>.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
1.1. Background	3
1.2. Introduction of Network Management Agent (NMA)	4
2. Terminology	5
2.1. Acronyms and Abbreviations	5
2.2. Definitions	5
3. Reference architecture of NMA	6
3.1. Intelligent Network Management and Control Framework Based on NMA	6
3.2. Function Requirements of NMA	10
3.3. Reference Architecture of NMA	11
3.4. NMA Interfaces	15
4. Deployment modes considerations and requirements	15
4.1. Single Agent Challenges	16
4.2. Multi Agents Challenges	17

5. Common processing flow of NMA	18
6. Typical Application Scenarios after Introducing NMA	19
7. Security Considerations	20
8. IANA Considerations	20
9. References	20
9.1. Normative References	20
9.2. Informative References	20
Authors' Addresses	22

1. Introduction

1.1. Background

As the types of operator services become increasingly diverse, the complexity and difficulty of network operations and maintenance continue to grow. On one hand, new service scenarios such as industrial internet, vehicle-road collaboration, and 5GtoB for vertical industries are constantly emerging, and customer services like Extended Reality (XR), Virtual Reality (VR), and smart home are becoming more abundant, with a continuous increase in network access volume. On the other hand, with the popularization of 5G and gigabit optical networks, operators' networks are facing a situation where networks from 2G to 5G coexist. The network protocols and characteristics vary across different network domains, leading to a continuous increase in the difficulty and complexity of network operations and maintenance. Relying solely on traditional manual operations and maintenance methods can no longer meet the increasingly complex network operations and maintenance demands. The level of network intelligence has become a key factor directly affecting network performance and user experience. Against this backdrop, enhancing the level of network intelligence and creating Autonomous Networks (AN)[TMF-IG1230] or Intent-based Networking [RFC9315] has become a global consensus among operators

Autonomous Networks provide an architecture for the delivery of services and capabilities with "Zero-X" (Zero-wait, Zero-trouble, Zero-touch) experience for the users of vertical industries and consumers and "Self-X" experience (Self-configuration, self-healing, self-optimizing) for network operators. In particular, the AN framework defines 6 automation levels, spanning from Level 0 (L0) where operations and maintenance are fully manual, to Level 5 (L5) where the network is fully automated, managed by the AI and the human intervention is reduced to the minimum.

As of today, the industry sees quite different levels of automation from operator to operator, but the average level is considered to be between L2 and L3. Mainstream operators are releasing goals and plans to achieve Level 4 (L4) autonomous networks by 2025. L4+ AN

sets higher requirement in intention, decision-making, analysis, perception, and execution. Artificial Intelligence (AI) large model technology has shown significant advantages and great potential in identification, understanding, decision-making, and generation. It has technical features such as multimodal fusion perception capabilities, more user-friendly human-computer interaction and knowledge Q&A capabilities, and content generation capabilities, which can well match the new requirements of Level 4 Autonomous Networks and already be one of the core driving technologies to achieve high-level autonomous networks.

While the key issues after the introduction of AI in network include:

- 1) The application architecture and deployment methods of AI in network management are still unclear, that is in what form AI can help network management?
- 2) The relationship between AI and the existing network controllers is not clear.
- 3) New interface capability requirements after AI is introduced are not clear either.

Therefore, it is necessary to define the general architecture and application form of AI in network management.

1.2. Introduction of Network Management Agent (NMA)

The concept of Network Management Agent (NMA) draws inspiration from the "AI Agent". According to the framework proposed in the blog[LLM-powered-autonomous-agents]by OpenAI's Lilian Weng, the functions of an LLM-powered Agent include several key components: planning, memory and using tools to complete actions. Following the mainstream definition widely accepted in the industry, an AI Agent refers to "an intelligent entity with the ability to perceive the environment, make decisions, and execute actions, and can gradually achieve set goals through independent thinking and tool invocation". In Google's latest Agent white paper[Agents], "a Generative AI agent can be defined as an application that attempts to achieve a goal by observing the world and acting upon it using the tools that it has at its disposal. Agents are autonomous and can act independently of human intervention, especially when provided with proper goals or objectives they are meant to achieve."

The key features of AI agents include reasoning and decision-making abilities, goal-orientation, and autonomy. Among these, autonomy means that once the appropriate goals are provided, it can act independently without human intervention. As the concept of AI agent becomes widely accepted in the industry, it's expected to become one of the most feasible application forms of AI.

Similarly, the network management agent (NMA) which can be understood as the AI Agent for network management, refers to a network management entity built based on ML/AI and equipped with the autonomous closed-loop task processing capabilities. It can automatically carry out network status perception, task intent interpretation, task planning, decision-making and task execution operations based on user task intentions or preset goals, so as to achieve closed-loop processing of scenarios-oriented network management tasks.

This document is trying to give a standardized common architecture for the use of AI in network management, which can be in the form of NMA. The following chapters will propose the concept of AI-based NMA, define the reference architecture of NMA and functional requirements of NMA for different scenarios, clarify the relationship of NMA with existing controller or other control systems, and discuss the general task processing workflow and typical application scenarios of NMA.

2. Terminology

2.1. Acronyms and Abbreviations

AI: Artificial Intelligence

LLM: Large Language Model

NMA: Network Management Agent, refers to AI based network management agent

2.2. Definitions

The document defines the following terms:

Network Management Agent (NMA): A network management entity built

based on ML/AI and equipped with the autonomous task processing capabilities. It can automatically carry out network status perception, task intent[RFC9315]interpretation, task planning, decision-making and task execution operations based on user task intentions or preset goals, so as to achieve closed-loop processing of scenarios-oriented network management tasks. For different application scenarios, NMA can be subdivided into multiple scenario-oriented agents.

3. Reference architecture of NMA

In this section we' ll analyze the functional requirements and reference architecture of the NMA.

3.1. Intelligent Network Management and Control Framework Based on NMA

As proposed in TMF' s AN journey guide : Autonomous Networks Level 4 industry blueprint high-value scenarios[TMF-AN-journey-guide], GenAI will be fully integrated into the AN three-layer architecture, i.e. resource operations layer, services operations layer and business operations layer, working in tandem with AI to develop two primary types of agent applications: role-oriented copilots and operation scenario-oriented agents. It should be noted that in this document, we will not make a distinction between the concepts of "copilot" and "agent"; instead, we will collectively refer to them as NMA. A copilot can be regarded as an initial-stage NMA or a specific type of NMA.

[RFC8969]proposed the framework for automating service and network management with YANG. Building on the architecture proposed in [RFC8969], higher-level intelligent network management and control can be achieved by adding NMA components. Based on the Figure 3 of[RFC8969], the layered architecture of intelligent network management and control after the introduction of NMA is shown in the following figure. NMA can exist at both the Controller and Orchestrator levels; for the device layer, due to the constraints on the computing power of network elements, some end-side AI components may be added on the device side, while it is unlikely to deploy a complete NMA.

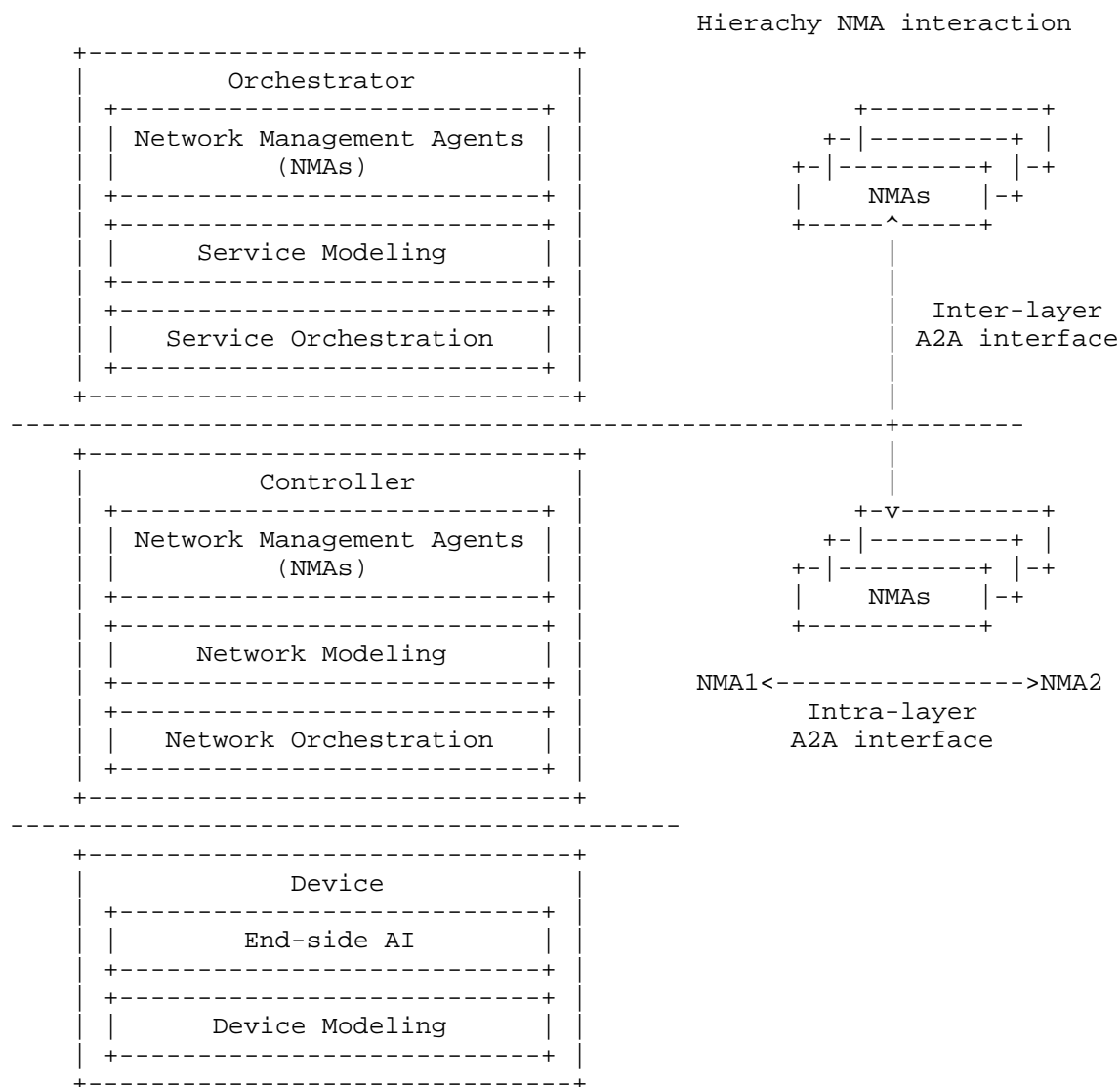


Figure 1: Enhanced intelligent network management and control framework based on NMA

Among them, there may be interaction requirements between NMAs at different layers and between different NMAs at the same layer. Cross-layer NMAs interact through inter-layer Agent-to-Agent (A2A) interfaces, while different NMAs within the same layer interact through intra-layer A2A interfaces. These two interfaces can adopt the same set of interface designs.

This document can be regarded as an enhancement of the intelligent capabilities of [RFC8969], and subsequent discussions will mainly focus on the NMA at the controller layer.

It should be noted that although NMA is depicted inside the controller and orchestrator in Figure-1, in practice, NMA can also be deployed as an independent component outside the controller and orchestrator. This document does not impose mandatory restrictions on the deployment location of NMA. The two deployment modes can be called: Independent deployment mode and Integrated deployment mode and are shown in Figure-2, where the NMA can be part of an existing network controller, or can be an independent system deployed separately and interacting both with the controller and the network.

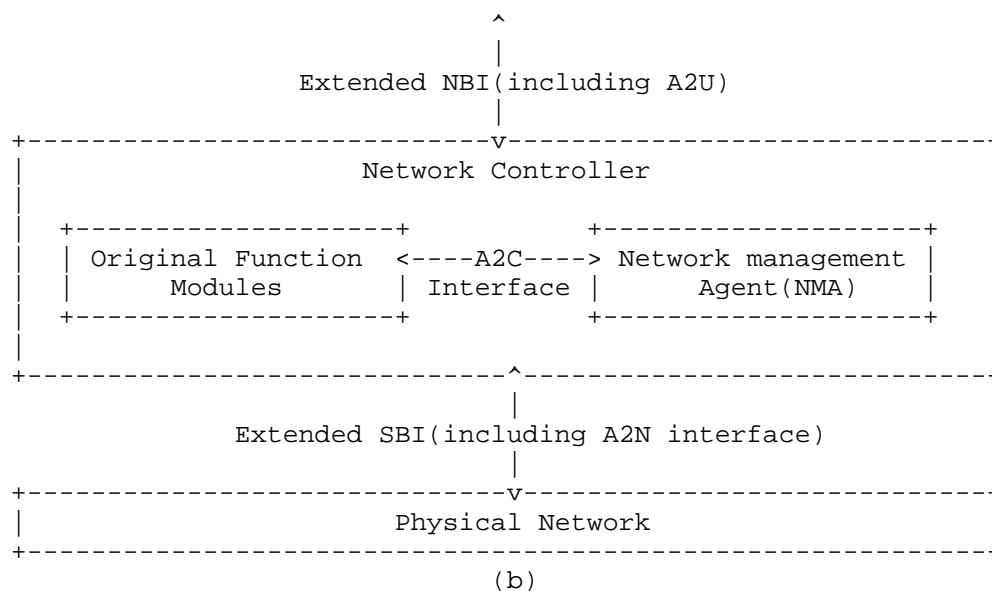
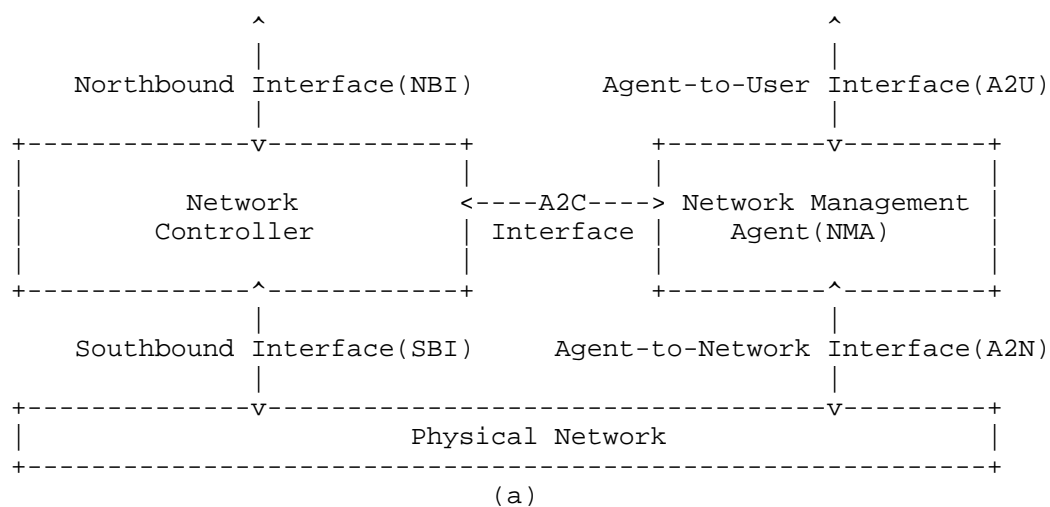


Figure 2: Deployment mode of network management agent (NMA)

Independent deployment mode: As shown in Figure 2(a), NMA is independently deployed from the original network controller. NMA and controller are independent systems. A new east-west interface needs to be added between the NMA and the controller to achieve capability calling and result feedback operations. This interface can be called “Agent-to-Controller Interface” (A2C). In this

deployment mode, controller uses southbound interface (SBI) to interact with physical network, while an Agent-to-Network interface (abbreviated as "A2N") needs to be added between NMA and the underlying physical network.

Integrated deployment mode: As shown in Figure-2 (b), NMA is integrated and deployed with the original network controller, and the NMA serves as a function of the controller. NMA interacts with original function modules through internal A2C interface. The enhanced controller interacts with the underlay physical network through extended SBI satisfying the A2N interaction requirements. The specific functional requirements and information model definition of interfaces mentioned above will be discussed in Section 3.4.

3.2. Function Requirements of NMA

The NMA should support the following capabilities:

1. Support receiving task requests initiated by network operators or users through natural language. It should be noted that natural language interaction is not the only way to use NMA, network operators can also use GUI (Graphical User Interface) to operate NMA. But NMA should have the capability of understanding natural language and translate into task intents through the build-in Large Language Models (LLMs) reasoning capability.
2. Support perception of network status through querying the data of controller and other network management tools. Network status include network topology, service configuration, alarms, performance and other information needed for processing the task.
3. Support task planning and breaking down task intent into specific operations based on the user input and network status perception. The task planning process can also utilize the reasoning capability of LLMs.
4. Support selecting appropriate tools and automatically invoking corresponding tools or APIs to complete the execution of each sub operation. The toolkit includes management functions from existing controller as well as other standalone management tools like Network Digital Twin (NDT) [I-D.irtf-nmrg-network-digital-twin], etc. Introducing new features in network is complex and extremely risky. A thorough service impact analysis must be performed before new features can be effectively enabled. Network digital twin can greatly help evaluate innovative network capabilities without jeopardizing the daily operation of actual network.

5. Support generating the task execution results based on the output of each operation and sending back to network operators or users.
6. Support analysis and self-assessment of execution results, and enable autonomous or human intervention optimization based on evaluation results to continuously improve the accuracy of task execution.
7. Support collaboration among multiple intelligent agents to complete complex tasks.
8. For operations that have a significant impact on the network, support simulation and verification before decision-making and delivery.

3.3. Reference Architecture of NMA

In order to achieve above capabilities, by referring to the common AI agent framework, this document presents the reference functional architecture of NMA as shown in Figure 1.

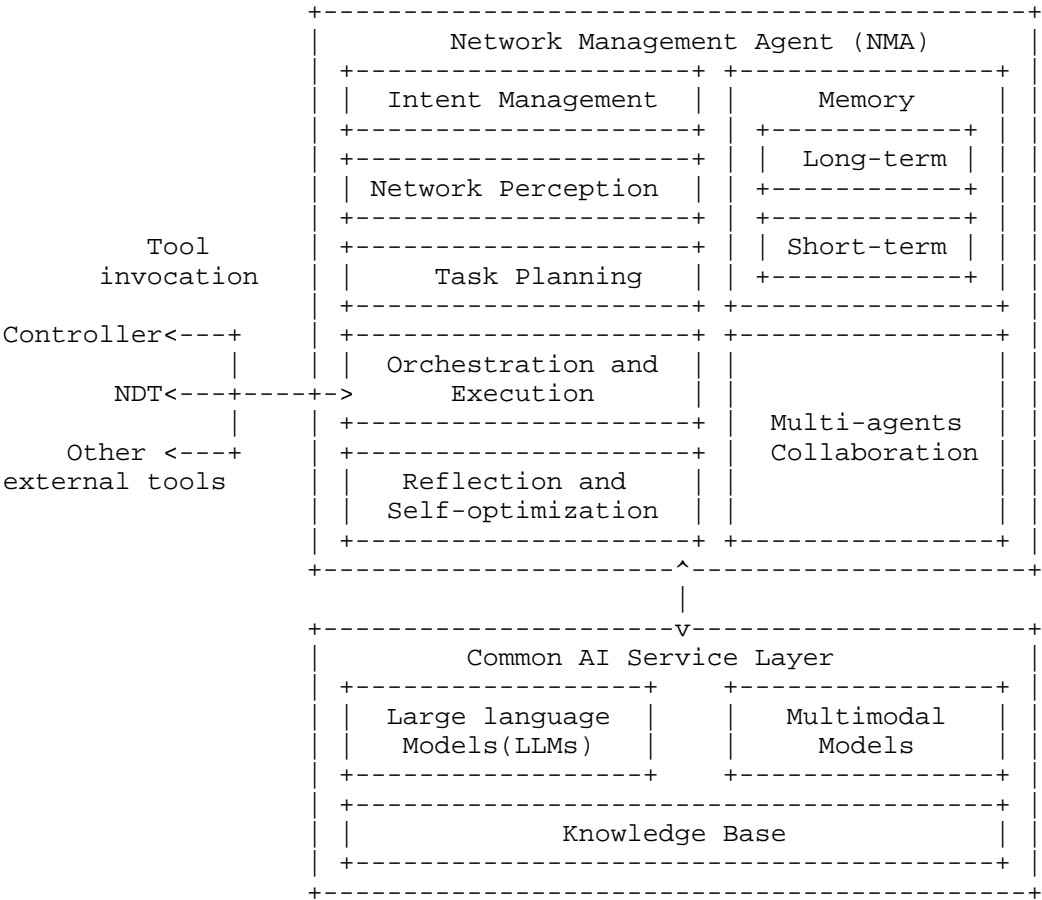


Figure 3: Reference function architecture of NMA

The main function components of NMA include:

Intent Management: Basic capability provided by AI models, responsible for collecting the input task information and translate into intents through AI model reasoning.

Network Perception: Achieve real-time query for network status information related to the task intent. Network status information is not limited to network topology, service configurations, device status, alarms, performances, etc. The query source can be controller, ENO, etc.

Task Planning: Based on the reasoning ability of AI models, break

down the task intention into multiple sub operations.

Orchestration and execution: Select the appropriate tools based on the specific operation, and automatically call the relevant tools or interfaces to perform the operation. After each sub operation is completed, the execution results of each operation are formed into task execution results.

Reflection and self-optimization: Select the appropriate tools based on the specific operation, and automatically call the relevant tools or interfaces to perform the operation. After each sub operation is completed, the execution results of each operation are formed into task execution results.

Additionally, artificial evaluation methods can be integrated to further optimize the NMA's performance through human supervision, enhancing the NMA's intention understanding and task execution capabilities.

Memory: Responsible for storing and processing various types of information during the operation of NMA, including long-term memory (LTM) and short-term memory (STM). STM stores information that NMA is currently aware of and needed to carry out complex cognitive tasks such as learning and reasoning. LTM can store information for a remarkably long time, ranging from a few days to months or years. To summarize, STM is for in-context learning which is short and finite, as it is restricted by the finite context window length of Transformer. LTM is for the external vector store that the NMA can attend to query time, accessible via fast retrieval.

Multi-agents collaboration Responsible for completing collaboration between multiple NMAs at different levels or in different application scenarios. The specific collaboration mechanism needs further research.

In addition, there is a common AI service layer, including various large language models (LLMs), multimodal models, small models, and knowledge base. Among them, AI models provide public interactive intelligence capabilities as unified agent engine, to simplify NMA development. Knowledge base provides unified search for multi-type knowledge bases including vector knowledge base, system online help, operation and maintenance data logs), combines AI models to complete knowledge fusion and extraction, and improves the accuracy of NMA task execution.

Various NMAs can be constructed based on the common AI service layer. During the operation of NMA, it leverages the model reasoning capabilities and knowledge base provided by the AI service layer to achieve functions such as intent parsing and task planning. It should be noted that, depending on the actual deployment requirements, the AI basic service can also be deployed within the NMA.

For different application scenarios, there can be multiple scenario-oriented agents (like apps in the phone). Aimed at the network planning, construction, maintenance, optimization, and operation scenarios, the main NMAs could include:

- * **Network Fault Handling Agent:** This agent can be created by pre-training specific AI model based on the network troubleshooting guidance documents, network equipment product documents, and other materials. The agent can solidify the fault handling experience of experts, and realize fault impact analysis, root cause self-diagnosis, and self-repair of network faults by orchestrating and calling models or network control APIs. It also interfaces with the work order dispatching system to achieve automated closed-loop processing of work orders, etc.
- * **Network Planning Agent:** Makes use of the capabilities of AI large model to understand the network planning intent (user intent, business development goals, network construction plans, etc.), and analyzes and forecasts the current network resource usage (traffic, performance, user scale, resource utilization, etc.) to output planning schemes.
- * **Network Optimization Agent:** Understands the network optimization goal through natural language, converts the optimization intent into network optimization constraint rules, such as network load thresholds, service route optimization strategies, etc. The instance can use traffic prediction models to predict the future traffic and bandwidth utilization of the entire network, automatically generate resource, hidden danger, performance, traffic, and other prediction results, and can automatically generate optimization strategies based on the prediction results to perform traffic pre-diversion, autonomous decision-making, and automatic execution to achieve dynamic energy saving of equipment and optimal traffic of the entire network, etc.

- * Intelligent Assistant Agent: This instance can have open Q&A capability based on LLM, providing a dialogue Q&A style operation and maintenance. Users can "one-click" input fault descriptions or resource names in natural language, and the instance will automatically perform intent recognition and query to significantly improve the efficiency of knowledge questioning, fault reporting, and maintenance support.

3.4. NMA Interfaces

As shown in Figure 2, the interfaces related to NMA include three types:

1. Agent-to-User interface (A2U): the interface between the NMA and the user, where the user can be a human or another system. This interface is used to receive call requests from users and return task processing results. It should support both structured and natural language modes. The natural language interface is mainly used for interaction with humans, while the structured interface is used for interaction with other upper-layer systems or other Agents. The Agent-to-Agent (A2A) interface between NMAs is included in the scope of this interface.
2. Agent-to-Controller interface (A2C): the interface between NMA and the controller or the original functional components of the controller. In the independent mode, this interface is an east-west interface between the controller and NMA; in the integrated mode, this interface is an internal interface of the controller and is not within the scope of this document.
3. Agent-to-Network (A2N): the interface between NMA and the physical network. In the independent mode, this interface is a southbound interface between the Agent and the network; in the integrated mode, this interface can be extended based on the original southbound interface of the controller.

The specific implementation methods, related protocols, etc. of each interface are to be defined subsequently.

4. Deployment modes considerations and requirements

While the integrated deployment mode is relatively simple, due to an internal communication between the NMA and the controller, the independent deployment mode introduces several challenges to be analyzed, that can be grouped into "single agent" and "multi agent" challenges.

4.1. Single Agent Challenges

Starting from an architecture with a single NMA, like the one shown in Figure 3 below, the challenges that we need to address are:

- * NMA APIs: Agents use descriptions of APIs and tools in order to use them. A gap analysis against existing tools needs to be carried out to understand if the NMA API requirements can be met and if we can find an optimal or common way to describe network APIs for LLMs.
- * NMA triggers: Agents need to be triggered with an input, which can be “just” a natural language input or something with a more structured format. Is the trigger going to be initiated by a controller or is it “just” a human readable string?
- * NMA interaction with existing controller: A wide variety of protocol and models exist today to interact with different components of existing controller. A gap analysis needs to be run to understand if those protocols and models are enough or extensions are needed in order to interact no longer with humans/UIs and higher order orchestrators/controllers but also by NMAs.

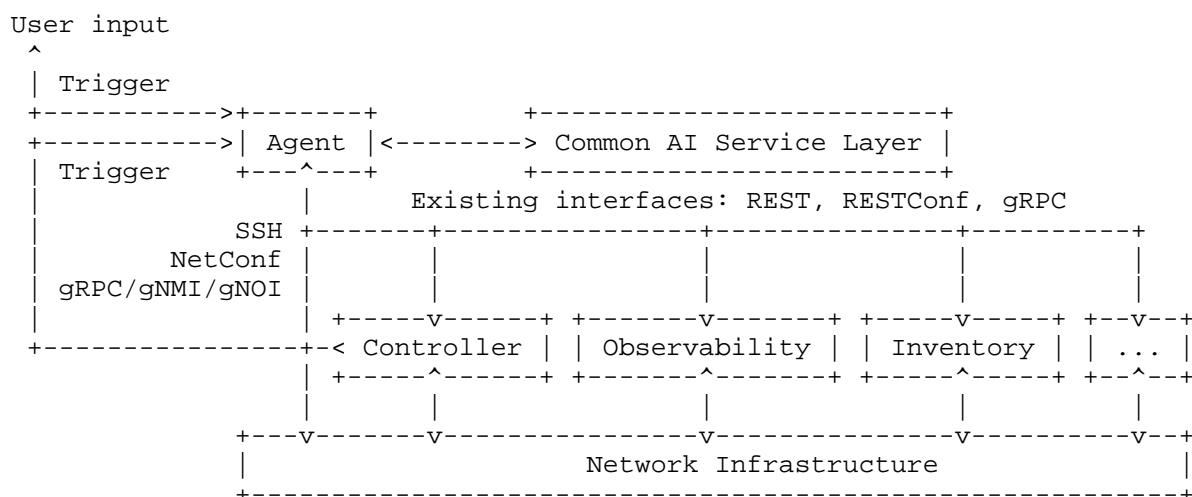


Figure 4: Network management architecture with single agent

4.2. Multi Agents Challenges

Things get a bit more complex when multiple NMAs are deployed, they need to interact with other NMAs as shown in Figure 4. In this case the challenges to consider are:

- * **Inter NMA communication:** The communication between agents can be achieved through a combination of natural language and structured data. Since this may involve the interaction between agents from different vendors or between operator and vendor, it's necessary to support authentication. Also, in multi-agent collaboration scenarios, there may be complex many-to-many interaction relationships between multiple Agents at the Controller layer and multiple Agents at the Orchestrator layer, resulting in intricate interaction dynamics. It is necessary to discuss whether to place all Agents in the Controller, with the Controller interacting with the Orchestrator in a one-to-one manner, or to add a many-to-many interface for Agents.
- * **NMA discovery and capabilities advertisement:** Since hierarchy management and control systems already deployed in the existing network, for a strongly planned network such as the transport network, the location of NMA is basically clear. However, even if discovery is not mandatory, capabilities advertisement between NMAs may still be required. How do we describe their capabilities? YANG based Restful interface may be a possible approach.

User input

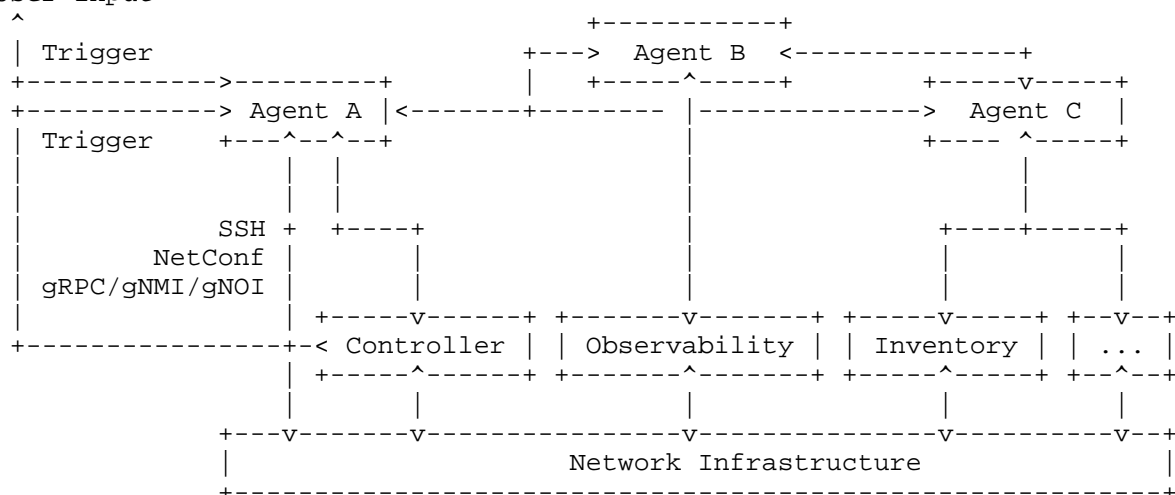


Figure 5: Network management architecture with multi agents

5. Common processing flow of NMA

The embedded AI model within NMA serves as the interface for user information input, and NMA instance uses the large model as the interface to clarify problems through multiple rounds, analyze positioning, generate plans, invoke interfaces/tools to handle problems, and complete closed-loop processing of problems, so as to build end-to-end problem processing assistance capabilities.

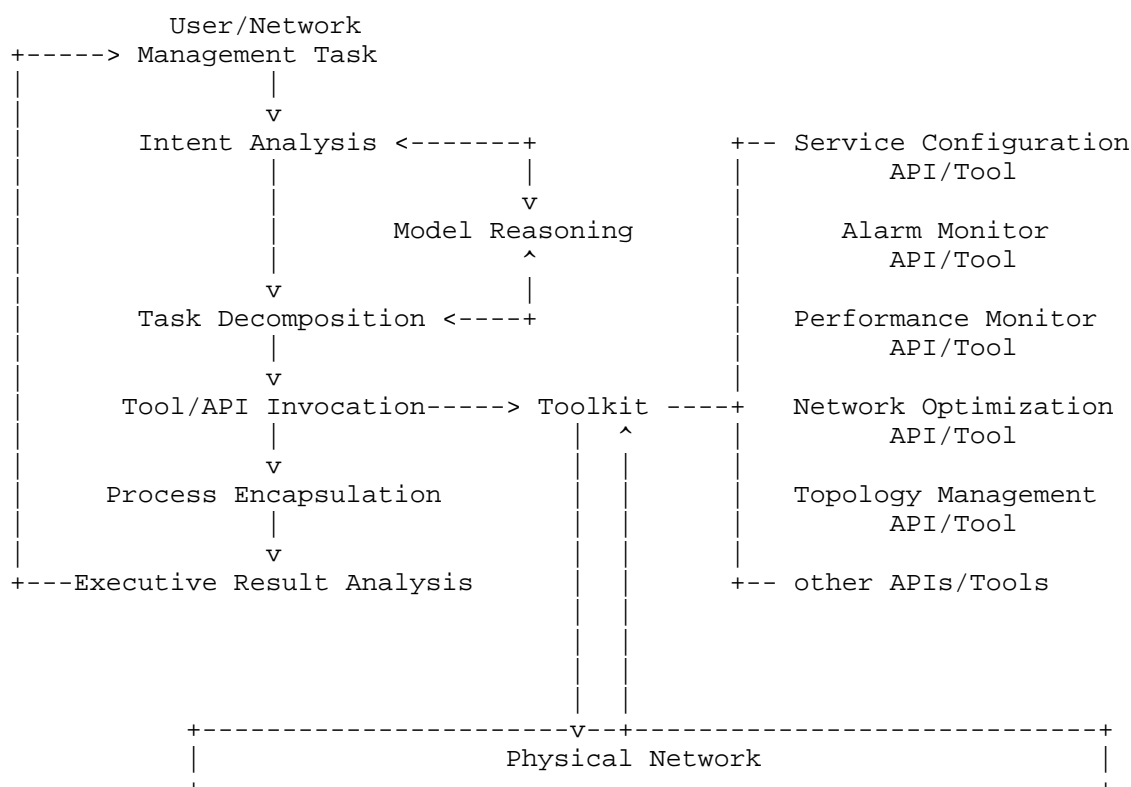


Figure 6: Common processing flow of NMA

The common processing flow of NMA instance are shown in Figure 3. The processing steps include:

1. User/Network Management Task Input: Input the user' s task information Through multiple rounds of natural language interaction.

2. Intent Analysis: Analysis user task intent through AI model reasoning provided by the AI based basic services within NMA.
3. Task Decomposition: Split the task into detailed operations to be performed based on the analyzed intent of the task.
4. Tool/API Invocation: Call the corresponding tool or function API to complete the execution of each operation listed in step 3). The toolkit refers to the collection of all tools that can be used directly to manage and operate physical networks, which can include management functions from existing controller, EMS, or standalone other management tools. The toolkit can include service configuration API/Tool, alarm monitor API/Tool, performance monitor API/Tool, network optimization API/Tool, topology management API/Tool, etc.
5. Process Encapsulation: Encapsulate each execution step. According to the order or dependency of all the operations, package the individual operation results into the execution result of the entire task.
6. Executive result analysis: Analyze the task processing results and return to the user.

Through above processing flow, NMA can achieve closed-loop automated processing of tasks and constructing end-to-end intelligent network maintenance assistance capabilities. For example, in the intelligent troubleshooting scenario, NMA can identify the cause of the fault and call the corresponding interfaces to handle it, such as creating a troubleshooting order, automatically initiating rerouting/optical power optimization, and other troubleshooting operations, and automatically verifying the progress of the order execution, with feedback on the troubleshooting results after the job order is completed.

The introduction of NMA can effectively improve the level of intelligent operation and maintenance of network, thus promoting the continuous evolution of communication network towards higher-level self-intelligence.

6. Typical Application Scenarios after Introducing NMA

Typical applications of NMA in networks can cover network operation and maintenance and operation processes:

Network management and maintenance scenarios, including:

- * Intelligent planning and construction: such as broadband installation, resource/capacity planning, intelligent acceptance, site selection, etc.
- * Intelligent maintenance: such as intelligent fault diagnosis, quality analysis, operation and maintenance/cutting assistant, broadband maintenance assistant, etc.
- * Intelligent optimization: such as route optimization, coverage optimization, topology optimization, and intelligent energy saving, etc.

Network operation scenarios: including intelligent question and answer, customer service assistant, automatic classification of user complaints, customer retention, product recommendation, automatic flow of work orders, anti-fraud monitoring and identification, intelligent marketing and other value-added services. This part is outside the scope of this document.

The starting point for the application of NMA in the live network should comprehensively consider the scenarios with strong demand, feasible technology, and good input-output ratio, and at the same time meet the requirements of sufficient data for AI pre-training during the construction of NMA instance, perfect data annotations, and high fault tolerance rate. Based on above considerations, the broadband installation and maintenance assistant, fault diagnosis, operation and maintenance assistant may become the first application scenarios.

7. Security Considerations

TBD.

8. IANA Considerations

This document has no requests for IANA action.

9. References

9.1. Normative References

9.2. Informative References

[Agents] Wiesinger, J., Marlow, P., and V. Vuskovic, "Google Whitepaper: Agents", 10 September 2024.

[I-D.irtf-nmrg-ai-challenges]

Franois, J., Clemm, A., Papadimitriou, D., Fernandes, S., and S. Schneider, "Research Challenges in Coupling Artificial Intelligence and Network Management", Work in Progress, Internet-Draft, draft-irtf-nmrg-ai-challenges-03, 4 March 2024, <<https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-ai-challenges-03>>.

[I-D.irtf-nmrg-network-digital-twin]

Zhou, C., Yang, H., Duan, X., Lopez, D., Paster, A., Wu, Q., Bouncaidair, M., and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture", Work in Progress, Internet-Draft, draft-irtf-nmrg-network-digital-twin-arch-09, 24 January 2025, <<https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-network-digital-twin-arch-09>>.

[I-D.kdj-nmrg-ibn-usecases]

Yao, K., Chen, D., Jeong, J., Wu, Q., Yang, C., and L. Contreras, "Use Cases and Practices for Intent-Based Networking", Work in Progress, Internet-Draft, draft-kdj-nmrg-ibn-usecases-01, 8 July 2024, <<https://datatracker.ietf.org/doc/html/draft-kdj-nmrg-ibn-usecases-01>>.

[LLM-powered-autonomous-agents]

Weng, L., "LLM Powered Autonomous Agents", 23 June 2023.

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

[RFC7576] Jiang, S., Carpenter, B., and M. Behringer, "General Gap Analysis for Autonomic Networking", RFC 7576, DOI 10.17487/RFC7576, June 2015, <<https://www.rfc-editor.org/rfc/rfc7576>>.

[RFC8969] Wu, Q., Boucadair, M., Lopez, D., Xie, C., and L. Geng, "A Framework for Automating Service and Network Management with YANG", RFC 8969, DOI 10.17487/RFC8969, January 2021, <<https://www.rfc-editor.org/rfc/rfc8969>>.

[RFC9222] Carpenter, B. E., Ciavaglia, L., Jiang, S., and P. Peloso, "Guidelines for Autonomic Service Agents", RFC 9222, DOI 10.17487/RFC9222, March 2022, <<https://www.rfc-editor.org/rfc/rfc9222>>.

[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/rfc/rfc9315>>.

[TMF-AN-journey-guide] Tansuthepverawongse, Boonchoung., "AN Journey Guide Autonomous Networks L4 industry blueprint-high-value scenarios", June 2024.

[TMF-IG1230] McDonnell, K., Machwe, A., Milham, D., O' Sullivan, J., Clemm, A., and J. Niemller, "Autonomous Networks Technical Architecture", TMF IG1230, December 2022.

Authors' Addresses

Xing Zhao
CAICT
Beijing
China
Email: zhaoxing@caict.ac.cn

Minxue Wang
China Mobile
Beijing
China
Email: wangminxue@chinamobile.com

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

Daniele Ceccarelli
Cisco
Email: dceccare@cisco.com

Jin Zhou
ZTE
China
Email: zhou.jin6@zte.com.cn

Haomian Zheng
Huawei
China
Email: zhenghaomian@huawei.com