Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

Y. Yang
Q. Wu
Huawei
D. Lopez
Telefonica
N. R. Moreno
Deutsche Telekom
L. Tailhardat
Orange ResearchAdd commentMore actions
H. Chihi
InnovCOM Sup'COM

7 July 2025

Applicability of A2A to the Network Management
draft-yang-nmrg-a2a-nm-00

Abstract

This document discusses the applicability of A2A to the network management in the multi-domain heterogeneous network environment that utilizes IETF technologies. It explores operational aspect, key components, generic workflow and deployment scenarios. The impact of integrating A2A into the network management system is also discussed.

About This Document

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

The latest revision of this draft can be found at https://Yuanyuan4666.github.io/A2A/draft-yang-a2a-nm.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-yang-nmrg-a2a-nm/.

Source for this draft and an issue tracker can be found at https://github.com/Yuanyuan4666/A2A.

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.

Table of Contents

1. Introduction

With the advancement of large language models (LLMs), the concept of AI agents has gradually attracted significant attention. An AI agent refers to a category of software applications that utilizes LLMs to interact with users or other agents and accomplish specific tasks. Take a multimodal AI agent as an example, it can collaborate with other domain-specific agents to complete diverse tasks such as translation, configuration generation, and API development.

A2A provides a standardized way for AI agents to communicate and collaborate across different platforms and frameworks through a structured process, regardless of their underlying technologies. Agents can advertise their capabilities using an 'Agent Card' in JSON format, or send messages to communicate context, replies, artifacts, or user instructions, which make it easier to build AI applications that can interact with heterogeneous AI ecosystems in specific domain.

With significant adoption of AI Agents across the Internet,Agent to Agent Communication protocol may become the foundation for the next wave of Internet communication technologies across domains [I-D.rosenberg-ai-protocols]. The application of A2A in the network management field is meant to develop various rich AI driven network applications, realize intent based networks management automation in the multi-vendor heterogeneous network environment. By establishing standard interfaces for dynamic Capability Discovery, intelligent message routing, heterogeneous AI ecosystems interaction,cross-platform collaboration, A2A enables AI Agents to:

o Understand contextual nuances

o Negotiate and adapt in real-time

o Make collaborative decisions

o Maintain persistent, intelligent interactions

This document discusses the applicability of A2A to the network management in the multi-domain heterogeneous network environment that utilizes IETF technologies. It explores operational aspect, key components, generic workflow and deployment scenarios. The impact of integrating A2A into the network management system is also discussed.

2. Conventions and Definitions

* AI Agent: A software system or program that is capable of autonomously performing goals and tasks on behalf of a user or another system.

* Agent Card: A common metadata file that describes an agent’s capabilities, skills, interface URLs, and authentication requirements. Clients discover and identify the agent through this file.

* A2A Server: An AI agent that receives requests and performs tasks

* A2A Client: An AI agent that sends requests to servers

3. Overview of key challenges for the network management

In large scale network management environment, a large number of devices from different network vendors need to be uniformly managed, especially in the heterogeneous network environment which can lead to the following issues:

3.1. Limitations of 3rd Party Management in Heterogeneous Network Environments

In the multi-vendor heterogeneous environment,vendors implementations of YANG models and NETCONF/RESTCONF protocols [RFC6241][RFC8040] exhibit significant divergence. Different vendors implement different YANG models such as IETF YANG, Openconfig YANG, Vendor specific YANG. Some vendors only partially support standard Network management protocols while Other vendors might choose non-stanard network management protocol or telemetry protocol such as gnmi [I-D.openconfig-rtgwg-gnmi-spec], grpc [I-D.kumar-rtgwg-grpc-protocol]. Without standard protocol or open programmable framework with multi-vendors integration drivers, integration various different data models and management protocols and allowing quickly adapt to different device are still big challenges. The same challenge is applied to multi-domain heterogeneous environment.

3.2. Static Data Format or Data Model for Management Interface, Unable to Adapt to the Speed of Service Roll Out

The IETF is currently working on and also publishing a set of YANG models for network service configuration. Network Service configurations are built from a combination of network element and protocol configuration, but are specified to service users in more abstract terms, which enables service agility to speed service creation and delivery and allows the deployment of innovative new services across networks. However Network service model provide static interface with a fixed, unchanging format, it is unable to adapt to new service requirements, e.g., when some new service attributes are introduced and correlated with the specific network service model A or knowledge graph B using RDF, it is hard to expose these new attributes or capability through the same management interface which is using network service model A.

3.3. YANG Model Lacks integration with Open APIs

Today, Open API has been widely adopted by the northbound interface of OSS/BSS or Network orchestrator while YANG data models have been widely adopted by the northbound interface of the network controller or the southbound interface of the network controller. However Open API ecosystem and YANG model ecosystem are both built as silo and lack integration or mapping between them.

4. Operational Consideration

This section outlines operational aspects of A2A with Network management requirements as follows:

* _Dynamic Capability Discovery and Negotiation_: Agent can automatically detect and understand each other's capabilities, enabling more intelligent and adaptive interactions, e.g.,client and remote agents can negotiate the correct format needed.

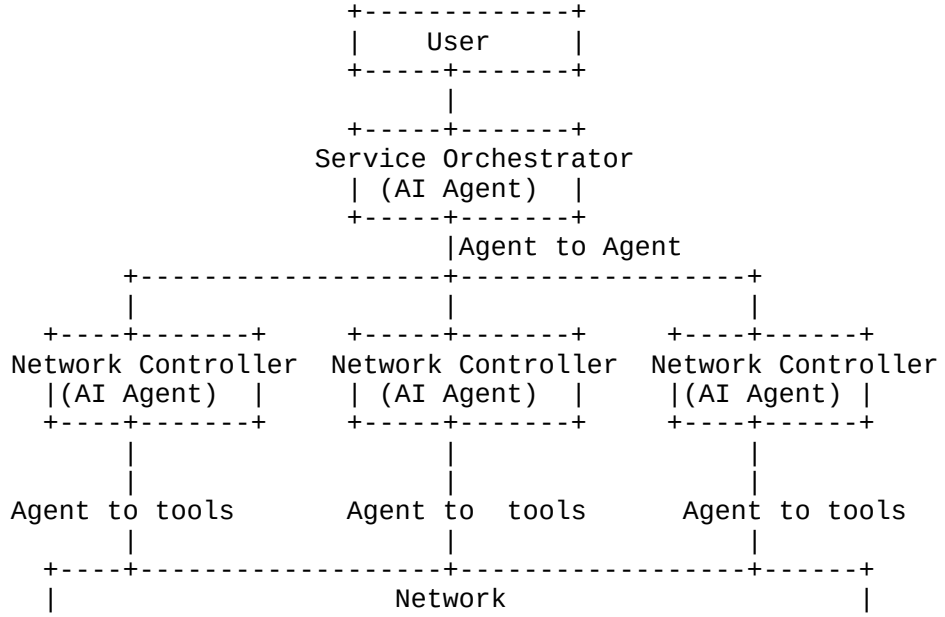
* _Task Management_: The communication between a client and remote agent is oriented towards task completion and agents work to fulfill end-user requests. The task object is defined by the protocol and has a lifecycle. Each of the agents can communicate to stay in sync with each other on the latest status of completing a task.

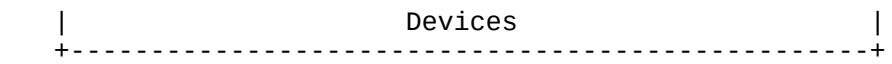
* _Automated Workflow Coordination_: Agents comprehend high-level user intent,execute extended workflow sequences. In addition, they enable more intelligent, context-aware agent interactions, e.g., Agents send each other messages to communicate context, replies, artifacts, or user instructions.

5. Architecture Overview

Large language models (LLMs) inherently excel at understanding complex user instructions, a capability that becomes even more pronounced in an AI agent-to-agent (A2A) architecture. Beyond merely comprehending sophisticated requirements, they can autonomously orchestrate lengthy network management workflows, making them particularly suitable for large-scale network management scenarios. Therefore, we have introduced the A2A protocol in the network management environments for building an intelligent network management and control platform.

5.1. Multi-Agent Communication Deployment Scenario





In the multi-agent communication deployment scenario, AI Agents can be deployed at both service layer and network layer,e.g., both service orchestrator and network controller can introduce AI Agent and allow Agent to Agent communication. AI Agent within the service orchestrator can provide registry database for other service agents within the network controller to register its location.

The interaction in the multi-agent communication deployment scenario can be break down into:

* _AI Agent to Agent interaction_

* _AI Agent to Tools interaction_

For AI Agent to Tools interaction, to enable comprehensive functionality, additional protocol extensions are required to address two critical aspects: (1) standardized tool invocation mechanisms for agent-tool interoperability, and (2) monitoring frameworks for tool usage tracking and auditing.

AI Agent to Agent interaction, users require a real-time monitoring interface for long-running workflows or tasks requiring continuous supervision with dual capabilities: (1) live network state observation and (2) validation of agent-proposed remediation actions during anomaly resolution scenarios.

A general workflow is as follows:

* User Input Submission: An operator submits a natural language request to a central AI agent.

* Agent Intent Processing: The central AI agent processes natural language inputs by parsing instructions into structured tasks.

* Workflow Graph Decision: The central AI agent decomposing tasks into workflow graphs, and distributes subtasks via an Agent Card Registry to specialized subordinate agents based on their capabilities.

* Iteration continues until all tasks reach executable leaf tier agents in the hierarchy.

* Leaf agents report outcomes to the central agent, which dynamically adjusts the workflow based on result analysis and policy rules.

5.2. Example

This section describes the deployment of a network configuration within a secure video meeting context. The scheduling agent is deployed to the Service Orchestrator, while the worker agent is deployed to the network controller. Registered on the Service Orchestrator, the agent card formally defines a worker agent’s capabilities, interfaces, and operational characteristics within network management systems.

See the following Agent card examples for two worker agents (QoS Agent and Security Agent):

```
 # Work Agents Capabilities
 {
     "name": "QoSAgent",
     "description": "Automatically configure QoS policies",
```

```
    "url": "https://qos-agent.example.com/tasks/send",
    "capabilities": ["QoS_Policy"],
    "skills": [
        {
            "id": "set_qos",
            "name": "QoS configuration",
            "description": "QoS configuration",
            "inputModes": ["text/structured"],
            "outputModes": ["text/status"]
        }
    ]
}
{
    "name": "SecurityAgent",
    "description": "Automatically configure network security policies",
    "url": "https://security-agent.example.com/tasks/send",
    "capabilities": ["IPSEC", "DTLS"],
    "skills": [
        {
            "id": "enable_encryption",
            "name": "Encryption method configuration",
            "description": "Encryption method configuration",
            "inputModes": ["text/structured"],
            "outputModes": ["text/status"]
        }
    ]
}
```

Suppose a user submits a natural language request such as "The meeting will have 100 participants. The security level is Top Secret" to the platform integrated with the Service Orchestrator. The platform parses the request and converts it into JSON format as follows:

```
# Requested Service Configuration
{
    "taskId": "task-multi-001",
    "action": "deploy_network_configuration",
    "parameters": {
        "context": "secure_video_meeting",
        "scope": "100",
        "secure_level": "Top Secret",
    }
}
```

The Service Orchestrator sends subtasks in a structured format to the Network Controller. For example, the subtasks for set_qos and enable_encryption are structured as follows:

```
# Set QoS and Enable Encryption Subtasks
{
    "taskId": "task-multi-001",
    "subTasks": [
        {
            "agent": "QoSAgent",
            "action": "set_qos",
            "parameters": {
                    "configuration": {
                            "acceptedOutputModes": [
                                    "text/status"
                    ]
                    },
                    "minimum_bandwidth": "100Mbps",
                    "priority": "0"
            }
        },
```

```
            {
                "agent": "SecurityAgent",
                "action": "enable_encryption",
                "parameters": {
                        "configuration": {
                                "acceptedOutputModes": [
                                        "text/status"
                        ]
                        },
                        "encryption_method": "ipsec",
                        "key_management": "dtls",
                }
            }
        ]
    }
```

The network controller executes network management operations on network devices and returns the results to the Service Orchestrator in JSON format. Example responses for the subtasks are shown below:

```
# Network Configuration Feedback Results
{
    "taskId": "task-multi-001",
    "action": "deploy_network_configuration",
    "parameters": {
        "context": "secure_video_meeting",
        "scope": "100",
        "secure_level": "Top Secret",
    }
}
{
  "taskId": "subtask-qos-001",
  "status": "completed",
  "artifacts": [{"type": "text", "content": "QoS setup completed"}]
}
{
  "taskId": "subtask-sec-001",
  "status": "completed",
  "artifacts": [{"type": "text", "content": "IPSEC encryption enabled"}]
}
```

6. Impact of integrating A2A on Network Management

6.1. Agent to Agent Interaction

A2A leverages advanced machine learning models or knowledge graph models and sophisticated communication protocols such as one built on top of HTTP, SSE, JSON-RPC.

Unlike REST or NETCONF/RESTCONF [RFC6241][RFC8040], other open API that follow predefined, static request-response patterns, A2A introduces a more adaptive communication model which transforms these interactions into dynamic, context-aware conversations.

In addition, Agents can now negotiate, interpret subtle contextual cues, and make collaborative decisions in real-time. The cost is more context information needs to be kept as states in both sides.

6.2. Agent to Tools Interaction

In case of collaboration between small AI model in the network element and large AI model in the network controller, A2A can be used to negotiate more context related information and invoke the tools. The cost is more context information needs to be kept as states in both sides.

In case of no lightweight AI in the network element, REST or NETCONF/ RESTCONF, other open API is sufficient for network management. There is no impact on management protocol used between the network element and the management system.

If YANG2CLI script has been deployed in the network element, this script can be used to translate YANG schema into CLI command and mange the 3rd party network device.

7. Security Considerations

The communication between Agents for the exchange of context information, capability information and user instruction is security sensitive and requires authentication,authorization and integrity protection. Legacy communication protocols such as HTTPS/TLS, designed for human-centric interactions, simply cannot withstand the high-speed exchanges between intelligent agents. Key security challenges in AI agent communication include:

* Identity Verification: Ensuring that agents are who they claim to be

* Data Integrity: Preventing unauthorized modifications during transmission

* Confidentiality: Protecting sensitive information from potential breaches

* Scalable Security: Maintaining robust protection across diverse and complex networks

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[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/rfc/rfc6241>.

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

9.2. Informative References

[I-D.kumar-rtgwg-grpc-protocol]
Kumar, A., Kolhe, J., Ghemawat, S., and L. Ryan, "gRPC Protocol", Work in Progress, Internet-Draft, draft-kumar-rtgwg-grpc-protocol-00, 8 July 2016, <https://datatracker.ietf.org/doc/html/draft-kumar-rtgwg-grpc-protocol-00>.

[I-D.openconfig-rtgwg-gnmi-spec]
Shakir, R., Shaikh, A., Borman, P., Hines, M., Lebsack, C., and C. Morrow, "gRPC Network Management Interface (gNMI)", Work in Progress, Internet-Draft, draft-openconfig-rtgwg-gnmi-spec-01, 5 March 2018, <https://datatracker.ietf.org/doc/html/draft-openconfig-rtgwg-gnmi-spec-01>.

[I-D.rosenberg-ai-protocols]

Rosenberg, J. and C. F. Jennings, "Framework, Use Cases and Requirements for AI Agent Protocols", Work in Progress, Internet-Draft, draft-rosenberg-ai-protocols-00, 5 May 2025, <https://datatracker.ietf.org/doc/html/draft-rosenberg-ai-protocols-00>.

Authors' Addresses

Yuanyuan Yang
Huawei
Email: yangyuanyuan55@huawei.com

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

Diego Lopez
Telefonica
Email: diego.r.lopez@telefonica.com

Nathalie Romo Moreno
Deutsche Telekom
Email: nathalie.romo-moreno@telekom.de

Lionel Tailhardat
Orange ResearchAdd commentMore actions
Email: lionel.tailhardat@orange.com

Houda Chihi
InnovCOM Sup'COM
Email: houda.chihi@supcom.tn