Network Working Group
Internet-Draft
Intended status: Informational
Expires: 6 May 2026

G. Zeng
J. Mao
B. Liu
N. Geng
X. Shang
Q. Gao
Z. Li
Huawei

2 November 2025

Gap Analysis of Network Configuration Protocols in LLM-Driven Intent-Based Networking
draft-zeng-opsawg-llm-netconf-gap-00

Abstract

Large Language Models (LLMs) are entering network operations through natural-language intent interfaces. Existing south-bound protocols (NETCONF, RESTCONF, gNMI, MCP, A2A) were not designed for conversational, semantically-rich, multi-agent orchestration. This document provides a systematic gap analysis and identifies extension points for each protocol to meet intent-based networking requirements.

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 6 May 2026.

Copyright Notice

Table of Contents

1. Introduction 2
2. Conventions and Definitions 3
3. Gap Analysis per Protocol 3
3.1. MCP 3
3.1.1. Gap Analysis 3
3.1.2. Solution Considerations 4
3.2. A2A 5
3.2.1. Gap Analysis 5
3.2.2. Solution Considerations 7
3.3. NETCONF 7
3.3.1. Gap Analysis 7
3.3.2. Solution Considerations 7
3.4. RESTCONF 7
3.4.1. Gap Analysis 7
3.4.2. Solution Considerations 8
3.5. gNMI 8
3.5.1. Gap Analysis 8
3.5.2. Solution Considerations 8
4. Summary 8
5. Normative References 8
6. Informative References 9
Authors' Addresses 9

1. Introduction

Intent-based networking (IBN) promises to translate high-level operator intent into network configuration without low-level syntax errors. With the advent of LLMs, the interface moves from YAML/CLI to natural language. Unfortunately, none of the current configuration or agent-to-agent protocols provide the semantic, transactional, and multi-agent primitives required by LLM-driven IBN. This draft analyses the gaps and proposes concrete extension directions for five widely deployed protocols.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", etc., are to be interpreted as described in [RFC2119].

IBN, LLM, Agent, Intent, Tool, Artifact, and Task are used as defined in [I-D.ietf-opsawg-ibn-terminology].

3. Gap Analysis per Protocol

3.1. MCP

3.1.1. Gap Analysis

The design goal of MCP is to give a single Large Language Model (LLM) a "plug-and-play" tool-calling capability. When deployed directly between a network controller and the devices, however, the following structural gaps are exposed.

3.1.1.1. Lack of Network-Level Transaction Semantics

MCP’s tools/call is a stateless, one-shot JSON-RPC invocation. Network changes normally require the multi-stage semantics "candidate → validate → commit → rollback." MCP has neither a candidate datastore nor two-phase-commit primitives. Consequently, cross-device bulk deployments cannot guarantee "all-or-nothing" atomicity. When partial failures occur, the controller must supply its own compensation logic, lengthening the LLM’s reasoning chain and increasing uncertainty.

3.1.1.2. No YANG Semantics Discovery Mechanism

Today MCP tool descriptors are written by hand. Network-device capabilities are authoritatively defined by YANG models; whenever a model is updated, the tool list must be manually re-synchronized. Without an automated pipeline "YANG → JSON-Schema → tool descriptor," maintaining the tool catalogue in a large multi-vendor environment becomes a bottleneck.

3.1.1.3. Encoding and Bandwidth Bottlenecks

Network-ops scenarios often involve high-frequency telemetry (560 s sampling, 10 k metrics per node). MCP specifies only JSON-RPC over HTTP/1.1, resulting in highly redundant messages and no streaming push primitive. When an LLM needs real-time anomaly detection, frequent polling consumes excessive bandwidth and CPU, violating data-center goals of low latency and high throughput.

3.1.1.4. Missing Multi-Device Context Correlation

MCP’s invocation context is confined to a single connection; it cannot natively carry network-level intent such as "change the same VLAN across three leaf switches while keeping the STP root bridge unchanged." The LLM must repeat the constraints in the prompt, wasting tokens and raising the error rate.

3.1.1.5. Lack of Network Rollback and Audit Hooks

Network operations require audit logs that "trace down to the leaf node." MCP’s tool return body contains only a JSON result; there are no standardized fields for rollback-point, commit-id, or syslog-severity. Root-cause analysis and compliance audits therefore require additional integration with device syslog or NETCONF logs, increasing cost.

3.1.1.6. Incompatibility with Existing Device Security Models

Devices commonly enforce certificate-based mutual-TLS plus NACM path-level permissions. MCP currently defines only a Bearer-token header and offers no mapping between a tool call and the read/write/exec permissions on a YANG node. If a tool-descriptor file leaks, an LLM could combine calls to bypass existing ACLs, creating a privilege-escalation risk.

3.1.1.7. Lifecycle and State-Management Gap

Network changes often last several minutes (waiting for BGP convergence or MAC migration). Once an MCP call completes, its context is discarded immediately, so there is no way to stream intermediate updates such as “convergence 90 %.” The LLM has no choice but to poll repeatedly, increasing load on both itself and the device while still failing to achieve a true state-machine-driven closed loop.

3.1.2. Solution Considerations

For MCP to serve as "the universal glue between LLMs and devices" in production networks, an upper layer must supply a transactional state machine, a YANG self-description channel, streaming encodings, and fine-grained audit semantics. Without these additions, MCP will remain confined to labs or single-device scripting and will be unable to close the loop on production-grade intent.

3.2. A2A

3.2.1. Gap Analysis

Positioned as a "multi-agent collaboration layer," the Agent-to-Agent (A2A) protocol was created so that any two LLM-Agents can discover each other, negotiate, and jointly finish long-running tasks. When it is dropped straight into "network-controller network-device" or "controller controller" settings, however, the following deep gaps surface:

3.2.1.1. Task Granularity Mismatch with Network Atomic Operations

A2A Tasks target macro-level business goals (e.g., "relocate a DC"). The smallest deliverable is an Artifact. Network changes, by contrast, must touch a single YANG leaf (e.g., "set interface X MTU = 9216"). The spec offers no "micro-task" primitive, so one Task either carries thousands of lines and becomes bloated, or is split into hundreds of Tasks that explode the state machine and raise LLM-orchestration complexity.

3.2.1.2. No Network-Wide Transaction or Roll-back Semantics

A2A’s state machine is limited to pending → working → completed/ failed. On failure the controller only gets a free-text Task.statusMessage. Network ops demand cross-device atomic commit plus a rollback tag. The protocol today defines no:

* two-phase-commit token (transaction-id),

* distributed lock or conflict detection,

* unified rollback API (rollback-on-failure).

Controllers must therefore implement compensation themselves, forcing the LLM to reason about "how to write a rollback script," which violates intent-based principles.

3.2.1.3. Poor Encoding and Bandwidth Efficiency

A2A mandates JSON for Artifact payloads and runs over HTTP/1.1. For high-frequency telemetry (5 s interval, 10 k metrics/node) or bulk config pushes, JSON’s textual redundancy causes:

* controller-device link congestion,

* wasted LLM-context tokens,

* repetitive header parsing and higher CPU load.

The protocol lacks a binary or streaming encoding option and offers no back-pressure mechanism.

3.2.1.4. Missing Multi-Device Context Correlation

A2A Task context is scoped to a single "conversation"; there is no standard field to express topology-level constraints such as "change the same VLAN on three leaf switches while keeping the STP root bridge unchanged." The LLM must repeat inter-device relations in the prompt, burning tokens and risking truncation that produces configurations which are syntactically valid but topologically wrong.

3.2.1.5. Incompatibility with Existing Device Security Models

Devices generally enforce certificate-based mutual TLS plus NACM path-level access control. A2A currently specifies only an OAuth2 delegation token and provides no mapping from "Task-level role" to YANG node read/write/exec permissions, nor per-Artifact fine-grained ACLs. Once an Artifact is cached or forwarded it may bypass the certificate chain, leading to privilege escalation or configuration pollution.

3.2.1.6. No Network-Semantics Discovery Mechanism

Skills are advertised in the Agent Card, but the Card is free text. There are no standard fields saying "I support OpenConfig BGP 4.0 YANG" or "I manage AS 65001-65500." LLMs must rely on fuzzy matching, often selecting the wrong partner and raising Task failure rates.

3.2.1.7. Life-Cycle and Intermediate-State Reporting Gap

Network changes can last minutes (waiting for BGP convergence, MAC moves). After a Task enters "working," A2A only mandates a final Artifact; there is no standard way to push interim states such as "convergence 70 %" or "MTU changed, waiting for LLDP neighbor re-discovery." The LLM must poll or wait until timeout, increasing load and preventing a true state-machine-driven closed loop.

3.2.2. Solution Considerations

To act as a "multi-agent collaboration bus" in network environments, A2A must be systematically extended in task granularity, transaction semantics, binary encoding, topology context, security mapping, life-cycle management, and intermediate-state push. Otherwise it will remain suited only for macro business flows and will be unable to close the fine-grained, reliable, and roll-backable network-intent loop required in production.

3.3. NETCONF

3.3.1. Gap Analysis

NETCONF [RFC6241] provides transactional, XML-encoded RPCs over SSH. It lacks:

* Semantic discovery: YANG models are not self-describing for LLMs; no runtime tool list.
* Session context: no standard place to store intent-id, LLM prompt, or multi-device correlation.
* Streaming telemetry: <notification> is push-style but insufficient for high-frequency KPI.
* Function-level audit: <commit> is atomic, but per-leaf authorization is out-of-scope.

3.3.2. Solution Considerations

TBD

3.4. RESTCONF

3.4.1. Gap Analysis

RESTCONF [RFC8040] maps YANG to HTTP URIs. Gaps include:

* No candidate datastore—every PUT/PATCH is immediate.
* No server-side discovery document for LLMs.
* Stateless: no place to store multi-request intent.
* Encoding flexibility may confuse LLM prompt consistency.

3.4.2. Solution Considerations

TBD

3.5. gNMI

3.5.1. Gap Analysis

gNMI delivers high-speed telemetry but:

* No semantic metadata for LLMs.
* Set() is non-transactional across multiple paths.
* No multi-agent signalling— gNMI is 1:1.
* No standardized error ontology.

3.5.2. Solution Considerations

TBD

4. Summary

No single protocol satisfies all IBN-LLM requirements. NETCONF/RESTCONF/gNMI need semantic and transactional extensions; MCP/A2A need networking-specific profiling. A companion document will define unified data models and security frameworks to close the identified gaps.

5. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", 1997.

[RFC6241] Enns, R., "Network Configuration Protocol (NETCONF)", 2011.

[RFC8040] Bierman, A., "RESTCONF Protocol", 2017.

[RFC9457] Nottingham, M., "Problem Details for HTTP APIs", 2023.

[OpenConfig-gNMI] Team, OpenConfig., "gNMI Specification", 2022.

[MCP-spec] Inc, Anthropic., "Model Context Protocol", 2024.

[A2A-spec] LLC, Google., "Agent-to-Agent Protocol", 2025.

6. Informative References

[I-D.ietf-opsawg-ibn-terminology]
IETF, "Intent-Based Networking Terminology", 2025.

Authors' Addresses

Guanming Zeng
Huawei
Email: zengguanming@huawei.com

Jianwei Mao
Huawei
Email: maojianwei@huawei.com

Bing Liu
Huawei
Email: leo.liubing@huawei.com

Nan Geng
Huawei
Email: gengnan@huawei.com

Xiaotong Shang
Huawei
Email: shangxiaotong@huawei.com

Qiangzhou Gao
Huawei
Email: gaoqiangzhou@huawei.com

Zhenbin Li
Huawei
Email: robinli314@163.com