Network Management Research Group
Internet-Draft
Intended status: Informational
Expires: 4 September 2025

Y. Cui
Tsinghua University
M. Xing
L. Zhang
Zhongguancun Laboratory

3 March 2025

A Framework for LLM-Assisted Network Management with Human-in-the-Loop
draft-irtf-nmrg-llm-nm-00

Abstract

This document defines an interoperable framework that facilitates collaborative network management between Large Language Models (LLMs) and human operators. The proposed framework introduces enhanced telemetry module, LLM decision module and standardized interaction data models between human operators and LLM-driven systems, and workflows to enforce human oversight. The approach ensures compatibility with existing network management systems and protocols while improving automation and decision-making capabilities in network operations.

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 4 September 2025.

Copyright Notice

Table of Contents

1. Introduction

1.1. Motivation

Traditional network automation systems struggle with handling unanticipated scenarios and managing complex multi-domain data dependencies. Large Language Models (LLMs) offer advanced multimodal data comprehension, adaptive reasoning, and generalization capabilities, making them a promising tool for network management and autonomous network[TM-IG1230]. However, full automation remains impractical due to risks such as model hallucination, operational errors, and the lack of accountability in decision-making[Huang25]. This document proposes a structured framework that integrates LLMs into network management through human-in-the-loop collaboration, leveraging their strengths while ensuring oversight, reliability, and operational safety.

1.2. Problem Statement

Network management faces significant challenges, including the complexity of multi-vendor configurations, the real-time correlation of heterogeneous telemetry data, and the need for rapid responses to dynamic security threats. LLMs offer a promising approach to addressing these challenges through their advanced multimodal data understanding and adaptive reasoning capabilities. However, the direct application of LLMs in network management introduces several technical considerations. These include the need for semantic enrichment of network telemetry to enhance LLM comprehension, a dual-channel decision execution mechanism with confidence-based escalation, and auditability of LLM-generated decisions through provenance tracking. Addressing these requirements is critical to integrating LLMs effectively into network management workflows while maintaining reliability, transparency, and interoperability.

2. Terminology

2.1. Acronyms and Abbreviations

* LLM: Large Language Model

* RAG: Retrieve Augmented Generation

3. Framework Overview

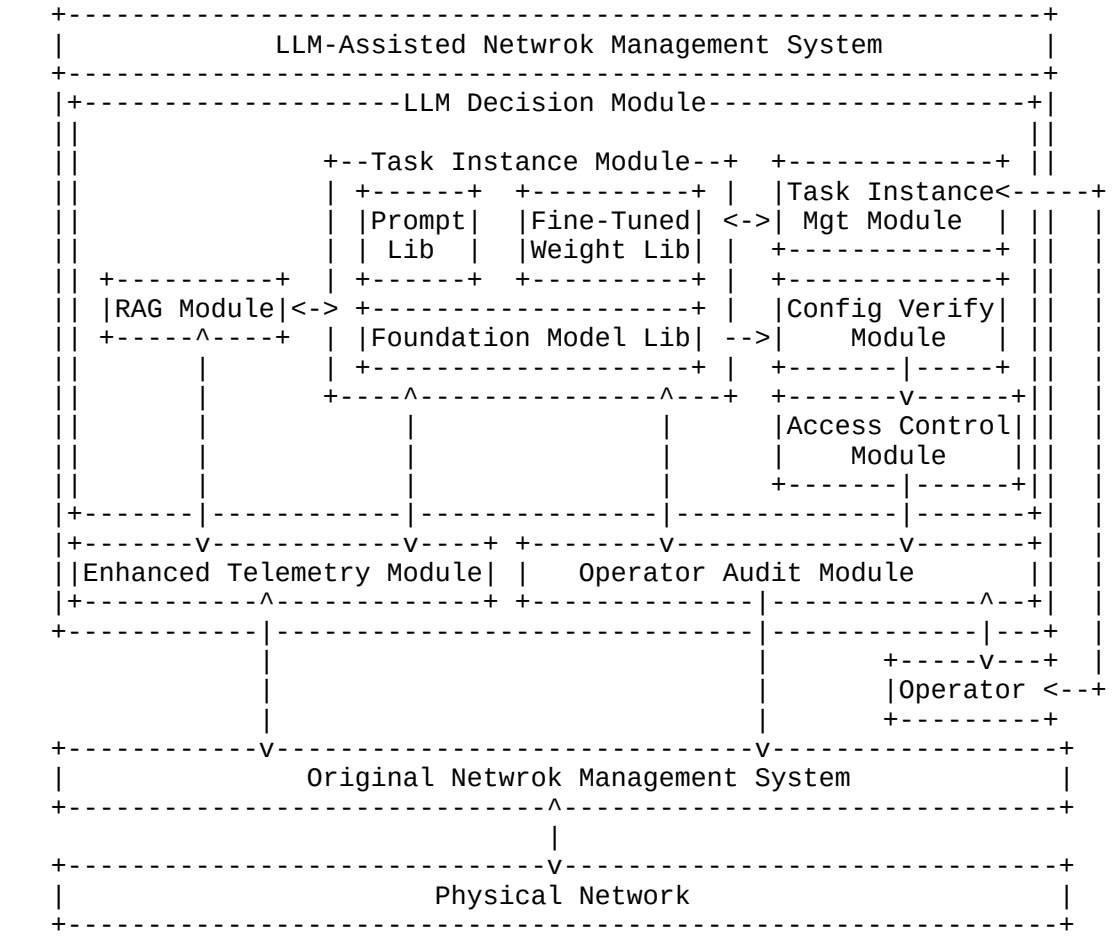

Figure 1: The LLM-Assisted Network Management Framework

The proposed framework is shown in Figure 1, highlighting the key components of LLM-assisted network management. The human operator can generate a specific task instance, e.g., fault analysis or topology optimization, using the task instance management module. According to the task type, the task instance module can instantiate a task instance with specific foundation model, prompt, and adaptation fine-tuned parameters[Hu22]. The enahnced telemetry module improves the semantics of raw telemetry data from original network management system, providing supplementary information to the LLM decision module for more informed decision-making. After the decision-making, the generated configuration parameters are validated against the YANG model and enforced with access control rules. The operator audit module provides a structured mechanism for human

oversight of LLM-generated configurations, and the configuration can be issued to the original network management system for deployment once the operator approves.

3.1. Enhanced Telemetry Module

The Enhanced Telemetry Module improves the semantics of raw telemetry data, providing supplementary information to the LLM decision module for more informed decision-making. Telemetry data retrieved from network devices via NETCONF[RFC6241], e.g., in XML format, often lacks field descriptions, structured metadata, and vendor-specific details. Since this information is not included in the pre-trained knowledge of LLMs, it can lead to misinterpretation and erroneous reasoning. To address this limitation, an external knowledge base should be introduced to store YANG model schema, device manuals, and other relevant documentation. The Enhanced Telemetry Module functions as middleware between the network management system and the external knowledge base. Through its southbound interface, it retrieves NETCONF data from the NETCONF client of existing network management system. Through its northbound interface, the module queries the external knowledge base for the corresponding YANG model or device manual. To enhance semantic richness, the Enhanced Telemetry Module processes the retrieved data by simplifying formatted content (e.g., removing redundant or closing XML tags) and appending XML path and description information from the YANG tree to the relevant fields. This approach ensures that the LLM has access to structured, contextually enriched data, improving its ability to analyze and reason about network telemetry.

3.2. LLM Decision Module

3.2.1. RAG Module

The pre-trained LLM may not encompass specific task requirements or vendor-specific knowledge. To address this kind of limitation, the Retrieve-Augmented Generation (RAG)[Lewis20] approach is widely used. This module retrieves relevant information from operator-defined sources, such as device documentation and expert knowledge, and integrates it with the Enhanced Telemetry Module to obtain YANG model schema. The retrieved textual data is stored in a database, either as raw text or in a vectorized format for efficient search and retrieval. For a given task context, the module retrieves relevant knowledge from the database and incorporates it into the input context, improving the understanding and response accuracy of LLM.

3.2.2. Task Instance Module

To execute a specific task, such as traffic analysis, traffic optimization, or fault remediation, a corresponding task instance must be created. A task instance consists of a selected LLM foundation model, an associated prompt and fine-tuned weights.

* Foundation Model Library. Operators must select an appropriate foundation model based on the specific task requirements. Examples include general-purpose models such as GPT-4, LLaMA, and DeepSeek, as well as domain-specific models fine-tuned on private datasets. Since foundation models are trained on different datasets using varying methodologies, their performance may differ across tasks.

* Fine-Tuned Weight Library. For domain-specific applications, fine-tuned weights can be applied on top of a foundation model to efficiently adapt it to private datasets. One commonly used approach is to store the fine-tuned weights as the difference between the original foundation model and the adapted model, which can largely reduce storage requirements. The Fine-Tuned Weights Module supports the selection and loading of an appropriate foundation model along with the corresponding fine-tuned weights, based on the selection of operators. This ensures flexibility in leveraging both general-purpose and domain-specific knowledge while maintaining computational efficiency.

* Prompt Library. For each task, it is essential to accurately define the task description, the format of its inputs and outputs. These definitions are stored in a structured prompt library. When an operator instantiates a task, the corresponding prompt, including placeholders for contextual information, is automatically retrieved. operator inputs and device data are then incorporated into the prompt at the designated placeholders, ensuring a structured and consistent interaction with the language model.

3.2.3. Task Instance Management Module

The Task Instance Management Module is responsible for the creation, update, and deletion of task instances. This module ensures that each instance is appropriately configured to align with the intended network management objective.

3.2.4. Config Verify Module

To ensure correctness and policy compliance, LLM-generated configurations MUST pass the YANG schema validation steps before being queued for human approval. This module ensures that only syntactically correct configurations are presented for operator review, thereby reducing errors and enhancing network reliability.

3.2.5. Access Control Module

Although the Configuration Verify Module can guarantee the syntactic correction, LLMs may generate unintended or potentially harmful operations on critical network devices, it is essential for operators to enforce clear permission boundaries for LLM task instance to ensure security and operational integrity. The Network Configuration Access Control Model defined in [RFC8341] provides a framework for specifying access permissions, which can be used to grant access to LLM task instances. This data model includes the concepts of users, groups, access operation types, and action types, which can be applied as follows:

* User and Group: Each task instance should be registered as a specific user, representing an entity with defined access permissions for particular devices. These permissions control the types of operations the LLM is authorized to perform. A task instance (i.e., user) is identified by a unique string within the system. Access control can also be applied at the group level, where a group consists of zero or more members, and a task instance can belong to multiple groups.

* Access Operation Types: These define the types of operations permitted, including create, read, update, delete, and execute. Each task instance may support different sets of operations depending on its assigned permissions.

* Action Types: These specify whether a given operation is permitted or denied. This mechanism determines whether an LLM request to perform an operation is allowed based on predefined access control rules.

* Rule List: A rule governs access control by specifying the content and operations a task instance is authorized to handle within the system.

This module must enforce explicit restrictions on the actions an LLM is permitted to perform, ensuring that network configurations remain secure and compliant with operational policies.

3.3. Operator Audit Module

The Operator Audit Module provides a structured mechanism for human oversight of LLM-generated configurations before deployment. The output from the LLM Decision Module should include both the generated configuration parameters and an associated confidence score. The configuration parameters are validated for compliance with the YANG model and are subject to Access Control Rules enforcement. The confidence score, e.g., ranging from 0 to 100, serves as a reference for operators to assess the reliability of the generated configuration. Each audit process must track the input context (e.g., input data, RAG query content, model selection, configuration files) and the corresponding output results. The auditing steps are as follows:

* Result Verification: The operator verifies the LLM-generated output to ensure alignment with business objectives and policy requirements.

* Compliance Check: The operator ensures that the LLM output adheres to regulatory standards and legal requirements.

* Security Verification: The operator examines the output for potential security risks, such as misconfigurations or vulnerabilities.

* Suggestions and Corrections: If issues are identified, the operator documents the findings and proposes corrective actions.

Upon completing the audit, the system maintains an audit decision record to ensure traceability of operator actions. The audit record includes the following information:

* Timestamp of the audit action

* LLM Task Instance ID associated with the action

* Operator decisions, including approval, rejection, modification, or pending status

* Executed command reflecting the final action taken

* Operation type (e.g., configuration update, deletion, or execution)

This structured approach ensures that all LLM-generated configurations undergo rigorous human review, maintaining operational accountability and security.

4. Data Model

This section defines the essential data models for LLM-assisted network management, including the LLM decision response and human audit records.

4.1. LLM Response Data Model

The LLM decision module should respond with the generated configuration parameters along with an associated confidence score. If the LLM is unable to generate a valid configuration, it should return an error message accompanied by an explanation of the issue.

```
module: llm-response-module
  +--rw llm-response
     +--rw config?         string
     +--rw confidence?     uint64
     +--rw error-reason?   enumeration
```

The LLM response YANG model is structured as follows:

```
module llm-response-module {
  namespace "urn:ietf:params:xml:ns:yang:ietf-nmrg-llmn4et";
  prefix llmresponse;
  container llm-response {
    leaf config {
      type string;
    }
    leaf confidence {
      type uint64;
    }
    leaf error-reason {
      type enumeration {
        enum unsupported-task;
        enum unsupported-vendor;
      }
    }
  }
}
```

4.2. Human Audit Data Model

This data model defines the structure for human audit operations and record-keeping. It facilitates collaborative decision-making by allowing LLMs to generate actionable insights while ensuring that human operators retain final operational authority.

```
module: human-audit-module
  +--rw human-audit
     +--rw task-id?      string
     +--rw generated-config?   string
     +--rw confidence?         int64
     +--rw human-actions
        +--rw operator?          string
        +--rw action?            enumeration
        +--rw modified-config?   string
        +--rw timestamp?         yang:date-and-time
```

The human audit YANG model is structured as follows:

```
module human-audit-module {
  namespace "urn:ietf:params:xml:ns:yang:ietf-nmrg-llmn4et";
  prefix llmaudit;
  import ietf-yang-types { prefix yang; }

  container human-audit {
    leaf task-id {
      type string;
      }
    leaf generated-config {
      type string;
      }
    leaf confidence {
      type int64;
      }
    container human-actions {
      leaf operator {
        type string;
        }
      leaf action {
        type enumeration {
          enum approve;
          enum modify;
          enum reject;
          }
        }
      leaf modified-config {
        type string;
        }
      leaf timestamp {
        type yang:date-and-time;
        }
    }
  }
}
```

5. IANA Considerations

This document includes no request to IANA.

6. Security Considerations

* Model Hallucination: A key challenge is that, without proper constraints, the LLM may produce malformed or invalid configurations. This issue can be mitigated using techniques such as Constrained Decoding, which enforces syntactic correctness by modeling the configuration syntax and restricting the output to conform to predefined rules during the generation process.

* Training Data Poisoning: LLMs can be trained on malicious or biased data, potentially leading to unintended behavior or security vulnerabilities. To mitigate this risk, LLMs should be trained on curated, high-quality datasets with rigorous validation and filtering processes. Periodic retraining and adversarial testing should also be conducted to detect and correct anomalies before deployment.

7. References

7.1. Normative References

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/rfc/rfc8341>.

[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>.

7.2. Informative References

[TM-IG1230] McDonnell, K., Machwe, A., Milham, D., O寰儡ullivan, J., Niemテカller, J., Varvello, L. F., Devadatta, V., Lei, W., Xu, W., Yuan, X., and Y. Stein, "Autonomous Networks Technical Architecture", February 2023.

[Huang25] Huang,, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., and T. Liu, "A Survey on Hallucination in Large Language Models Principles, Taxonomy, Challenges, and Open Questions", n.d..

[Hu22] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and W. Chen, "LoRA Low-Rank Adaptation of Large Language Models", n.d..

[Lewis20] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kテシttler, H., Lewis, M., Yih, W.-t., Rocktテ、schel, T., and S. Riede, "Retrieval-augmented generation for knowledge-intensive NLP tasks", n.d..

Authors' Addresses

Yong Cui
Tsinghua University
Beijing, 100084
China
Email: cuiyong@tsinghua.edu.cn
URI: http://www.cuiyong.net/

Mingzhe Xing
Zhongguancun Laboratory
Beijing, 100094
China
Email: xingmz@zgclab.edu.cn

Lei Zhang
Zhongguancun Laboratory
Beijing, 100094
China
Email: zhanglei@zgclab.edu.cn