2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) | 979-8-3503-8053-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/NFV-SDN61811.2024.10807498

An Agentic Approach For Dynamic Software-Defined Network Management Using Large Language Models

1st Aramis Sales Araujo
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
aramisaraujo@copin.ufcg.edu.br

2nd Jefferson Maxmiliano O. das Mercês
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
jefferson.merces@embedded.ufcg.edu.br

3rd Rilbert Lima da Silva
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
rilbert.silva@embedded.ufcg.edu.br

4th Allender Vilar de Alencar
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
allender.alencar@virtus.ufcg.edu.br

5th Iele Facundo Passos
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
iele.passos@virtus.ufcg.edu.br

8th Thiago Fonseca Meneses
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
thiago.meneses@virtus.ufcg.edu.br

6th Marcelo Portela Sousa
Federal Institute of Paraíba
Campina Grande, Paraíba, Brazil
marcelo.portela@ifpb.edu.br

7th Michel Coura Dias
Federal Institute of Paraíba
João Pessoa, Paraíba, Brazil
michel.dias@ifpb.edu.br

9th Danilo F. S. Santos
Virtus RDI Center
Federal University of Campina Grande
Campina Grande, Paraíba, Brazil
danilo.santos@virtus.ufcg.edu.br

***Abstract*—In recent years, Large Language Models (LLMs) have been in the spotlight over very diverse fields of research and development. Powered by the concept of Agents, LLMs have acquired substantial tools to perceive and act upon the environment they are applied to. Even though a few works have explored the application of LLMs in the field of dynamic management of network environments, most stop at the application of this technology at Natural Language Processing tasks such as identifying intents, leaving the interpretation and execution of those to humans. This work proposes the application of a reflexive architecture for LLM-powered Agents, enabling perception of network and application data and aiming to autonomously act upon a Software-Defined Network (SDN) environment. The problem of dynamic SDN management and application-aware optimizations is tackled by the novel approach in this work. An experiment was conducted with the proposed Reflexive Agent acting upon a simulated 5G Core Network and video streaming application, through which early results show that the applied LLM-powered Agent approach was able to exert precise actions, achieving the defined intents and providing comprehensive reasoning to support its actions.**

***Index Terms*—Large Language Model, Agent, Intent-Based Network, Software-Defined Network, Quality of Service, 5G Mobile Network**

I. Introduction

The emergence of 5G mobile networks has revolutionized the telecommunication industry, providing unprecedented bandwidth, ultra-low latency, and massive connectivity [1]. Those advancements paved the way for innovative applications and services across different areas including autonomous vehicles, smart cities, and industrial automation.

One of the critical challenges in these network environments, however, is ensuring that their services meet specific application Quality of Service criteria, such as low latency or high throughput for data-intensive applications [2]. Traditional network management approaches, which rely on static configurations and manual interventions, are costly from an operational point of view as they rely on active human intervention, unlike autonomous paradigms that may leverage that cost [3].

The exploration of autonomous approaches has led to the integration of Artificial Intelligence (AI) driven approaches such as Intent-Based Networking (IBN), which enables networks to automatically adjust their behavior based on intents defined at a higher level language, achieving dynamic objectives defined by a network manager or stakeholders [2]. Taking the spotlight, AI and Machine Learning (ML) tools have emerged as powerful components for enhancing network automation and programmability. By leveraging ML techniques, for example, mobile networks can dynamically adapt to unexpected events, optimize resource allocation, and proactively address potential issues [4].

This work proposes that a dynamic network management system can be achieved through monitoring systems and agents. To reach this goal, an agentic architecture has been employed through which Large Language Models (LLMs)

This work has been partially funded by FINEP/Brazil grant number Ref. 2826/22 with financial resources from FNDCT/MCTI.

empower Agents not only to perceive information about a 5G mobile network and applications running within, but also to act, meeting intents defined in natural language.

Observing this scenario, the definition of agents may be twofold:

i) In the network view, it represents the means through which the network core configures and interacts with physical devices;
ii) However, in the Artificial Intelligence (AI) view, it refers to the broader and abstract concept of Intelligent Agents, as in autonomous entities that leverage the power of AI to perform various tasks.

The referred Agents are able to make complex decisions, learn, adapt, and correct themselves, thus excelling in the execution of tasks in diverse environments.

In further details, the proposed architecture focuses on the employment of a reflexive and agentic approach through which multiple agent tools are able to act upon the transport network and applications by:

i) Querying metrics;
ii) Proposing actions;
iii) And executing those actions to meet the defined goals.

This agentic and reflexive architecture also enables the provision of reasoning for performing the actions, thereby effectively acting upon the network environment.

These characteristics allow for optimization opportunities in the supporting models, such as fine-tuning with supervised learning, multi-agent resources and a broader set of tools for the model to act with. These opportunities promise improvements not only in response time but, more importantly, in accuracy when interacting with the core network environment to meet the defined intents.

In summary, this work aims to contribute to the field of dynamic management of software-defined networks by introducing an architecture through which Agents powered by Large Language Models perceive and act upon the network. Leveraging their programmability aspects and effectively complying with intents defined in natural language autonomously.

This article is structured as follows: Section II discusses the use of AI and ML in enhancing network automation and how IBN differs from traditional network models. In Section III the architecture of the experiment is detailed, including the simulation setup for a 5G network within PNETLab [5] and the role of Reflexive Agents acting upon the ONOS controller [6] for network optimizations. Section IV outlines preliminary results, demonstrating the Agents' ability to discern network situations and autonomously select suitable tools. This work conjectures a discussion on the outcomes and future research directions.

II. Related Work

The adoption of 5G networks and developments of what is to be Beyond 5G technologies heighten the need for efficient network management and orchestration solutions. Key concepts such as Intent-Based Networking (IBN) and other Artificial Intelligence Driven network management techniques focus on enhancing Quality of Service and network programmability. In this scenario, IBN represents a paradigm shift from traditional and software-defined networking by introducing a higher level of automation and intelligence in network management.

According to Fan's study on Intent-Based Networking (IBN), traditional networks rely heavily on manual configuration, device-specific commands and are prone to human error and inefficiency [7]. Implementing IBN also allows network administrators to define the desired goals (intents) in a natural language level, bridging complex business intents and network management solutions. An IBN system uses Artificial Intelligence and Machine Learning techniques to analyze, determine and execute the necessary actions to meet the defined goals, thereby enhancing operational efficiency and reducing the potential for errors [7].

The integration of Artificial Intelligence (AI) into network management, particularly in the context of 5G and Beyond 5G, has been explored extensively. Abbas et al. propose an IBN system to automate network slicing and ensure intelligent resource allocation through the Network Data Analytics Function [8]. The proposed system employs a hybrid stacking ensemble learning algorithm to predict network resource utilization and leverages an automated machine learning approach for anomaly detection. This AI-enhanced architecture significantly improves the accuracy of resource predictions and anomaly detection, as demonstrated by their experimental results showing a 20% increase in accuracy and a 45% reduction in error rate [8].

Another critical development is the application of Natural Language Processing (NLP) in network management. Mcnamara et al. discuss using NLP to facilitate user interactions with network management systems, allowing non-expert users to specify intents using natural language commands [9]. This approach simplifies interactions and leverages advanced Artificial Intelligence techniques to translate natural language goals into actionable network configurations. The study highlight the potential of NLP to bridge the gap between business objectives and technical implementations, making network management more accessible and efficient [9].

Emerging from the evolution of Natural Language Processing techniques and the combination of different model architectures such as Transformers [10], Large Language Models are powerful, interactive and intelligent tools that are built upon vast knowledge bases which suggest an opportunity to apply them in the autonomous network management scenario [11].

Large Language Models (LLM) alone, however, often face constraints when applied to an autonomous application scenario. One of the most critical limitations is the length of context that an LLM can support at a given time, the capacity is directly tied to the training regimen undergone by the models and the underlying hardware capabilities. Therefore, this becomes a limiting factor for storing large amounts of context data needed for standalone applications. Another important

limitation is the lack of direct tool utilization capabilities as these models are not comprised of means through which they can directly interact with the environment [12].

Precisely approaching these limitations, the deployment of Agents enable Large Language Models to perceive the context they are applied to and empower them with a set of tools to effectively interact and modify the environment. Cheng et al. highlight that agents generally exhibit the following characteristics:

- **Autonomy**: Agents independently perceive their environment, make decisions, and take actions without relying on external instructions;
- **Perception**: Agents are provided with the capability of gathering information about the environment through sensors;
- **Decision making**: Agents make decisions based on perceived information by analyzing and formulating actions to meet their defined goals;
- **Action**: Agents perform actions that alter the state of their environment.

Going further, the application of Reflexive Agents [13] provide pondering and refining capabilities to traditional agents, enabling precise decision making. These characteristics are vital for the interaction of agents with critical scenarios, as it helps mitigate hallucination effects [14] from the language models and refines proposed actions, as portrayed in the diagram of Figure 1, by:

1) Interpreting a user's prompt or request;
2) Providing a response to that user's prompt;
3) Executing a tool from the provided tool set;
4) Reflecting upon the proposed response, refining the answer by removing superfluous information or inputting missing details;
5) Providing a final refined answer.

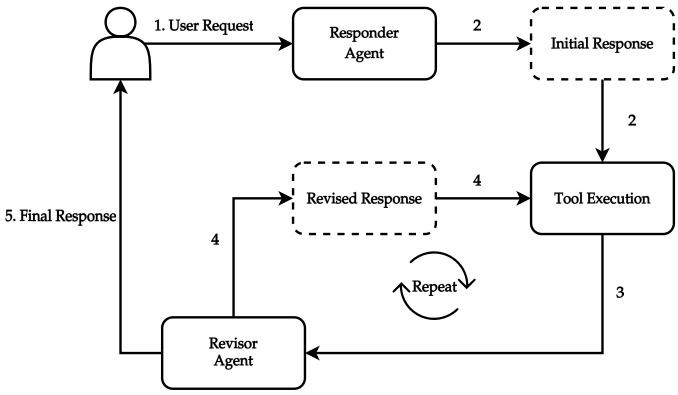

Figure 1. Reflexive Agent Architecture Diagram.

In conclusion, integrating Artificial Intelligence techniques into network management promises significant progress such as higher efficiency, scalability, and alignment with business objectives [4]. These technologies are poised to revolutionize the field of network management, particularly with the transition to more complex and dynamic environments like Beyond 5G.

III. Architecture and Experiment Design

In order to evaluate the potential of a reflexive agent architecture in the task of network management, an experiment was devised as follows: A 5G network was deployed within a PNETLab [5] environment, using Free5GC [15] as a network core simulator and UERANSIM [16] as User Equipment (UE) and gNodeB simulated components. Within this network environment, simulated UE were used to benchmark the network capabilities through the deployment of a video streaming application.

A reflexive agent application was developed using LangChain [17] and LangGraph [18] Python libraries and given control of this network environment through ONOS' API [6]. Through the API, the agent was able to deploy different flow configurations, thereby modifying the network.

The diagram in Figure 2 depicts the simulation setup of the 5G network architecture using PNETLab, a network emulation platform tailored to create, test and simulate network scenarios [5]. With this platform, various network devices such as switches, routers, and firewalls can be emulated, allowing for topology testing and scenario simulation in a controlled environment.

Within a PNETLab instance, two Free5GC [15] Virtual Machines (VM) were independently executed. In the VM designated as Control Plane (Figure 2), components such as the Access and Mobility Management Function (AMF) and Session Management Function (SMF) were deployed.

The AMF is responsible for managing authentication, registration, connections, and mobility of devices within the 5G network. The SMF, on the other hand, manages user data sessions in the 5G network based on the service requested by the User Equipment (UE). Another VM was used for deploying the User-Plane Function (UPF), which is responsible for transporting IP data traffic between the UE and external networks. In this experiment's context, the UPF is directly connected to a Media Server application, allowing for the recreation of a distributed computing environment within the 5G network, as displayed in Figure 2.

In order to simulate different User Equipment (UE), UERANSIM [16] was employed as a component that emulates both UE and gNodeB (gNB). Virtual machines running within the PNETLab environment were deployed for this particular component. In this specific experiment, five VMs were used for simulating UEs, and two VMs for simulating the gNBs, corresponding to "gNB-1" and "gNB-2" (Figure 2).

This architecture was employed to facilitate the capture of packets for analysis. It is important to note that, by default, UERANSIM can simulate both the UE and the gNB on a single VM, however, for this use case, the division of these elements into different VMs allowed for improved analysis and understanding of the system's operation. This structure also allowed the recreation of the dynamic connection seen on a Radio Access Network component. Specific deployment details of the components are displayed in Table I.

The proposed experiment architecture, depicted in the diagram of Figure 3, consists of a video streaming application

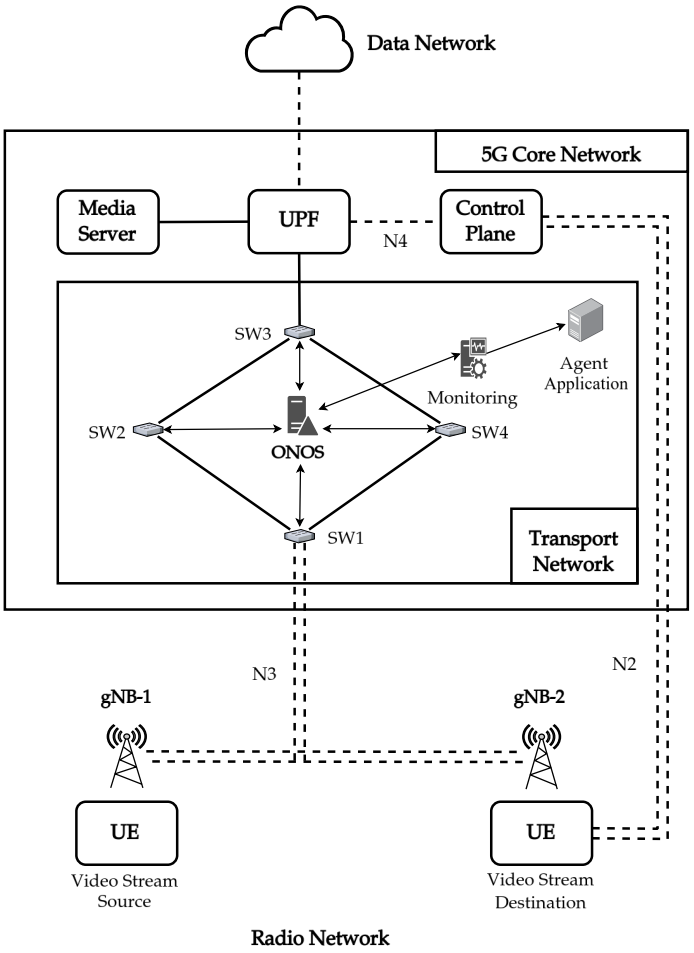

Figure 2. Network and Experiment Diagram.

TABLE I
TECHNICAL DETAILS OF THE SETUP

Emulated within PNETLab		
Component	**Instances**	**Characteristics**
Transport Network (Open vSwitch)	4	4 vCPUs & 4 GB RAM
5G Core (free5GC)	1	4 vCPUs & 4 GB RAM
Simulated gNodeB (UERANSIM)	2	4 vCPUs & 4 GB RAM
UPF (free5GC)	1	4 vCPUs & 4 GB RAM
UEs (UERANSIM)	5	8 vCPU & 16 GB RAM
Media Server Application	1	16 vCPU & 16 GB RAM
ONOS	1	4 vCPU & 4 GB RAM

running on a 5G core network environment. The performance metrics for both network and applications are collected and stored in a time-series database, allowing for tracking of the

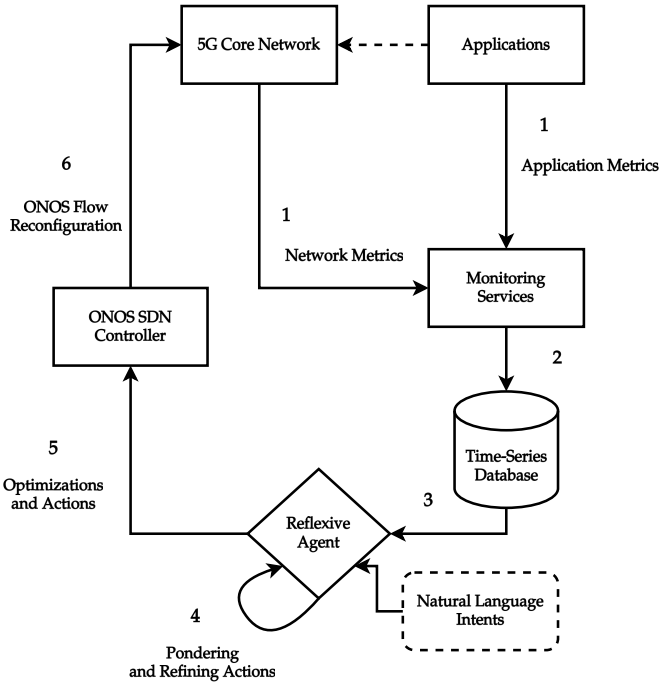

Figure 3. Architecture of a reflexive agent acting upon a 5G core network and applications.

system's status promptly. A Reflexive Agent analyzes these data records, guided by natural language intents defined as prompts to act upon the ONOS Controller. The agent is able to deploy different ONOS flows, thus re-configuring the network. This setup enabled continuous monitoring, analysis, and optimization of the 5G core network and applications.

The agentic application was developed using Meta's LLaMA 3.0 model [19] without any further fine-tunning on the scenario or experiment data. This model was used to perform the actions depicted in the diagram of Figure 3 through the following steps:

1) First, application and network core related metrics are scraped by the monitoring system;
2) The scraped metrics are then stored in a time-series database, tracking their time-sensitive aspects that depict the system's status at a given point in time;
3) The Reflexive Agent implementation utilizes the available tools to query the application and network data, generating an initial response given the intents provided in natural language;
4) The initial response and actions are then analyzed by the agent's revisor component, refined through pondering iterations by removing superfluous actions and missing details are then included in it;
5) Finally, the Agent utilizes the provided tools to apply different flow configurations at the ONOS Controller, effectively re-configuring the network (6).

IV. Preliminary Results and Discussion

The first steps in the experiments required introducing semantic meaning for the Agent through the use of prompts. The following prompts were used for this purpose:

- A prompt for the **Responder Agent component**, which is responsible for setting up which tools it has access to, as described in Listing 1.
- A prompt for the **Reviser Agent component**, which is responsible to review previous answers, as described in Listing 2.
- An initial prompt from a human user, describing the expected goals (intents), as seen in Listing 3.

```
You are an expert network manager that controls network
fluxes using ONOS.

You have access to a time−series database containing data
about the current network status and application services.

Your first task is to fetch the latest data from the database
using the "fetch_data" tool which will yield tabular data.
Then, decide whether to apply any of the following tools:

  i) Apply improvement flow for latency;
  ii) Apply improvement flow for packet drop;
  iii) Apply improvement flow for throughput;
  iv) Remove a flow.

If critique is provided, you may need to apply different tools to
achieve the results based on the data provided and your
previous actions.
```

Listing 1. Prompt provided to Responder Agent

```
Reflect upon an answer.

Provide critiques upon what is missing and what is superfluous.

Revise your previous answers using the acquired information
and previous critiques to add important information to the final
answer.

You must include reasoning in your revised answer to ensure it
can be verified and use previous critiques to remove superfluous
information.
```

Listing 2. Prompt provided to Reviser Agent (reflexion)

```
Ensure the applications are running with low latency, enough
throughput and a low rate of packet drop.
Provide reasoning for the actions you take to achieve that, based
on the data you retrieve.
```

Listing 3. Initial Prompt provided by the User

After the initial system configuration, experiments in the simulated network environment were performed, focusing on the video streaming application that was running on the simulated 5G network, as described in Section III.

During the execution of the application, different network scenarios were simulated to evaluate the following conditions:

- High latency;
- High rate of packet drop;
- Low throughput.

With these network scenario simulations, it was observed whether the proposed solution was capable of performing the following steps:

1) Identify network scenarios;
2) Identify the appropriate tool to apply;
3) Autonomously apply the tool.

Preliminary results show that the model supporting the Agents was able to succeed in all these steps, as follows:

1) Identified the application context through the semantics of the fields in the database, e.g. *'rtsp_bytes_sent'* relates to a video streaming application, as presented by the agent response in Listing 4;
2) Understood the semantics of each provided tool, specifically the functions for applying different flows at the ONOS SDN controller;
3) Used the provided '*fetch_data*' tool to fetch time-series data related to the application and network;
4) Understood the semantics of the queried data and identified the proper tool to solve the introduced latency scenario, as presented by the agent response in Listing 5.
5) Properly applied a predefined ONOS flow, by using a provided tool for re-routing the application through different one, effectively leading to lower latency in the streaming application, as presented by the agent response in Listing 6;
6) Provided proper reasoning for the application of the ONOS flow (among the other available options), thus reducing the latency in the application and achieving the previously defined goals.

```
The network and application data indicates that the application
is experiencing high latency and a high rate of packet drop.

These issues are likely caused by network congestion or outdated
equipment.

Currently, the network is experiencing a high rate of packet loss,
which is affecting the quality of video streaming.
```

Listing 4. Identification of network scenarios by Responder Agent

```
Based on the original question, the first goal is to keep the
latency low.

However, the current issue is with packet loss, which can impact
the quality of video streaming.

To address this issue, I have decided to apply the "Apply
improvement flow for packet drop" tool as it can help increase
the network's capacity and prevent future packet loss issues.
```

Listing 5. Identification of possible tools to apply by the Agent

```
Based on the selected tool, the solution was applied using
ONOS controller's REST API to reconfigure the network.
```

Listing 6. Result of the application of tools

V. Conclusion

In conclusion, the proposed approach introduces a novel Agentic Approach that leverages LLM-powered agents for the autonomous management of Software-Defined Network environments. The deployed approach seems particularly adept at application-aware optimizations, as showcased by a noteworthy experiment within a simulated 5G Network context, paired with a video streaming application, displaying the agent's capability to select targeted actions and articulating their rationale.

The preliminary results are promising, suggesting that the LLM-powered agent methodology is effective at manipulating the network to fulfill specific objectives, thereby supporting the potential for dynamic Software-Defined Network management. Furthermore, a reflexive approach using the LLM-as-Agent to monitor, act and re-configure the network was able to pursue the defined goals for Quality of Service related metrics.

As future work, comparative benchmarks across various LLM models could be executed with a semantic and qualitative approach. Other important improvement points are the fine-tunning of the Large Language Models on network-specific data and the exploration of dynamic network slice manipulation within 5G Network configurations. These directions not only pave the way for advancing the field but also highlight the transformative impact of LLMs in network management.

Acknowledgment

This work has been partially funded by FINEP/Brazil grant number Ref. 2826/22 with financial resources from FNDCT/MCTI, and supported by CENTRO DE COMPETÊNCIA EMBRAPII VIRTUS EM HARDWARE INTELIGENTE PARA INDÚSTRIA - VIRTUS-CC, with financial resources from the PPI HardwareBR of the MCTI grant number 055/2023 signed with EMBRAPII.

References

[1] M. Ramachandran, T. Archana, V. Deepika *et al.*, "5g network management system with machine learning based analytics," *IEEE Access*, vol. 10, pp. 73 610–73 622, 2022.

[2] A. Kousaridas, R. P. Manjunath, J. Perdomo *et al.*, "Qos prediction for 5g connected and automated driving," *IEEE Communications Magazine*, vol. 59, no. 9, pp. 58–64, 2021.

[3] D. M. Manias, A. Chouman, and A. Shami, "Towards intent-based network management: Large language models for intent extraction in 5g core networks," in *2024 20th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 2024, pp. 1–6.

[4] Y. Ahn and J. P. Jeong, "An intent-driven management automation for 5g mobile networks," in *2024 International Conference on Information Networking (ICOIN)*, 2024, pp. 714–719.

[5] "PNetLab," https://pnetlab.com/, accessed: 2024-07-18.

[6] "ONOS - Open Network Operating System," https://opennetworking.org/onos/, accessed: 2024-07-11.

[7] L. Fan, "A study of intent-based networking," *University of Alberta Library*, 2023. [Online]. Available: https://era.library.ualberta.ca/items/3b2df1a2-8055-41c6-a735-db66642a5569

[8] K. Abbas, A. Nauman, M. Bilal *et al.*, "Ai-driven data analytics and intent-based networking for orchestration and control of b5g consumer electronics services," *IEEE Transactions on Consumer Electronics*, 2023.

[9] J. Mcnamara, D. Camps-Mur, M. Goodarzi *et al.*, "Nlp powered intent based network management for private 5g networks," *IEEE Access*, vol. 11, pp. 36 642–36 657, 2023.

[10] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] C. Wang, M. Scazzariello, A. Farshin *et al.*, "Netconfeval: Can llms facilitate network configuration?" *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT2, pp. 1–25, 2024.

[12] Y. Cheng, C. Zhang, Z. Zhang *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects," *arXiv preprint arXiv:2401.03428*, 2024.

[13] N. Shinn, F. Cassano, A. Gopinath *et al.*, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] Z. Ji, T. Yu, Y. Xu *et al.*, "Towards mitigating llm hallucination via self reflection," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 1827–1843.

[15] C. Chang, F.-C. Chen, J.-C. Chen *et al.*, "free5gc," https://github.com/free5gc/free5gc, 2021, accessed: 2024-07-18.

[16] "UERANSIM," https://github.com/aligungr/UERANSIM, accessed: 2024-07-18.

[17] H. Chase, "Langchain," https://github.com/langchain-ai/langchain, 2022, accessed: 2024-07-18.

[18] "Langgraph," https://github.com/langchain-ai/langgraph, 2024, accessed: 2024-07-18.

[19] A. Dubey, A. Jauhri, A. Pandey *et al.*, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783