

# AI-Generated Network Configurations for Software-Defined Networking (SDN)

Tushar Mohan Sawant<sup>1</sup>, Aarti Sanjay Gawai<sup>2</sup>

<sup>1</sup>Student, Department of Information Technology and Computer Science, D.G. Ruparel College, Mumbai, Maharashtra, India

<sup>2</sup>Assistant Professor, Department of Information Technology and Computer Science, D.G. Ruparel College, Mumbai, Maharashtra, India

<sup>1</sup>[sawanttush385@gmail.com](mailto:sawanttush385@gmail.com), <sup>2</sup>[aartigawai1611@gmail.com](mailto:aartigawai1611@gmail.com)

\*\*\*

**Abstract** - In this paper, we review recent work using AI to generate configurations for Software-Defined Networking (SDN). This is not a straightforward task, as SDN enables network management to be “programmed” on paper, but actual configuration of network devices remains a challenging, error-prone task. High-level intent needs to be translated to low-level device settings. We present a range of recent approaches, building on top of Large Language Models (LLMs), Generative AI, Intent-Based Networking (IBN), and multi-agent systems, and in particular on top of Retrieval-Augmented Generation (RAG). We assessed and compared the performance of the current state-of-the-art AI-based tools for generating network configurations, including the recently developed framework for generating SDN configurations based on Intent decomposition, Contextual retrieval, and IaC (Infrastructure as Code) for provisioning, testing and deployment, especially for SDN and cloud-native networks. We go into great detail and present an in-depth analysis of the current work and especially the shortcomings, such as the lack of formal verification, the risk of vendor lock-in, security risks, lack of explainability, and lack of learning in a dynamic network. Our results show that the currently available AI-generated configurations for networks hold promise for enabling fully self-managing, scalable, and dynamic networks. However, several validation tools and more concrete proof-of-work need to be delivered by industry before these technologies can be deployed in production.

**Key words:** retrieval-augmented generation (RAG), network automation, intent-based networking (IBN), large language models (LLMs), and software-defined networking (SDN).

## 1. INTRODUCTION

Today, almost all organizations use computer networks for communication, data sharing, online services and many other activities. As a network grows in size ,

keeping track of all devices and their configurations can be a tedious job . Network administrators frequently have to configure different devices and ensure they work well together. This can take a long time, especially if changes are needed frequently. Software Defined Networking provides an alternative way to manage networks. It decouples the control of the network from the physical devices and it enables many network operations to be controlled by software. Therefore, SDN is useful to create networks that are easier to control and automate. But the administrators should understand the network needs, and translate them into the and translate them into the policies, rules, and configurations that the software controller can enforce across the network infrastructure.

Recent upgrades in AI, in particular generative AI and large language models (LLMs), offer new opportunities to simplify network management. Instead of having to write each configuration manually, administrators can simply state what they want the network to do, and artificial intelligence can create a configuration based on those instructions. This can cut down time and repetitive work involved in network configuration. This study looks at how AI can be used to generate network configurations in Software-Defined Networking (SDN) environments.

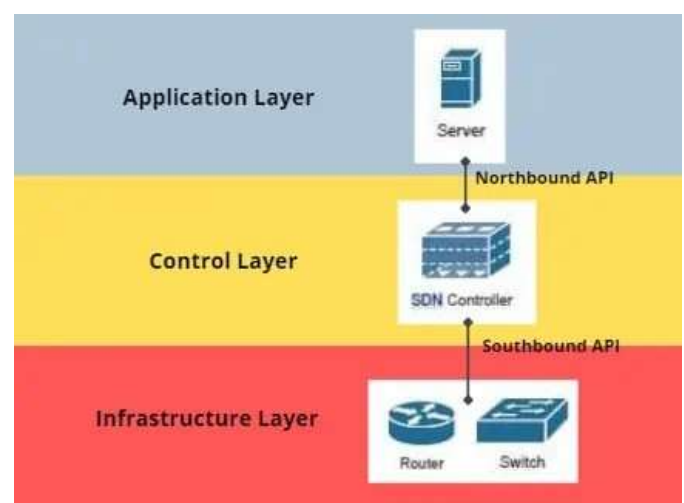


Fig -1: Referenced from study-ccna.com

## 2. Review of literature

Network management has evolved far from the days of strictly manual infrastructure, thanks largely to recent advances in Software-Defined Networking (SDN) and intent-based automation. But let's face it, even with a centralized SDN controller providing good programmability humans still are stuck doing the tedious work. Admins still need to take high-level business requirements and map it manually into exact low-level device configurations. This is a large operational bottleneck. Which is why the recent arrival of Large Language Models (LLMs), Generative AI, feels like such a true watershed moment. Those tools are finally starting to eat away that brutal daily operational complexity, allowing engineers to just write out requirements in plain, natural language and get instant ready-to-run network configurations in return. In their work, Brodimas et al. (2025) suggest an Intent-Based Infrastructure and Service Orchestration framework based on an Agentic AI. The presented framework integrates IBN, LLMs, multi-agents, and IaC to make the process of managing networks and infrastructure much easier. Instead of working according to some predefined workflows, the solution provides users with an ability to define their needs using plain language. The Intent Decomposition Agent (IDA) is used in the framework to manage different specialized agents and to interpret the needs. The agents are able to perform various actions related to infrastructure provisioning, Kubernetes cluster management, and service deployment. This way, it is possible to achieve higher levels of automation, however, human intervention is still required for critical actions.

One major aspect of the contribution made by this study lies in the use of a distributed multi-agent reasoning approach in place of a monolithic AI-based reasoning approach. Every agent has an individual responsibility, and some of the responsibilities include analyzing intent, providing the required infrastructure, managing resources, and deploying applications. The reasoning approach that is used in this case is ReAct (Reasoning and Acting). It allows agents to analyze user intent, call the required tools, interact with other agents, and create Infrastructure as Code configurations. This framework further incorporates Retrieval-Augmented Generation (RAG) and Model Context Protocol (MCP) for adequate context understanding as opposed to simply relying on

pre-trained knowledge for configuration generation. Hence, prior to any configuration generation, it retrieves relevant domain information and corresponding infrastructure information, and subsequently generates appropriate deployment plans. By storing both long-term memory and short-term memory, this framework facilitates consistent multi-step reasoning throughout complex network orchestration steps. The integrated framework with support for Infrastructure-as-Code (IaC) tools. The framework utilizes IaC to interpret user intent and provision the necessary cloud based infrastructure. This enables configuration of Kubernetes clusters as well as the subsequent deployment of network services. In contrast to SDN controllers which typically provide users with interfaces to interact with the system, such as a REST API or a graphical user interface, the framework under consideration translates natural language statements into actions to deploy required network services and corresponding configuration.

Experimental evaluation of the Agentic AI framework proves significant increase in completion rates of tasks, accuracy of configurations and in efficiency of tasks compared to manual approaches of orchestrating same tasks. In particular the framework was tested by setting up an Open5GS-based 5G core network and a Kubernetes cluster, and then using the framework for fully automated provision of necessary infrastructure for running the core network, with minimal to no human interaction, and with strict validation to guarantee reliable execution of the tasks. The results from this research are comparable to recent research on Leveraging Generative AI for Intent-Based Networking Operations in Network Slices. The recent research involved the use of LLMs integrated into an SDN controller in order to Automatically Interpret User Intent, Validate Network Parameters, Automatically Generate Corresponding Network Configuration, and Explain the Corresponding Intent(s). Results from the recent research were able to achieve Factual Accuracy of around 93% while Reducing Manual Configuration Effort. The recent research as well as the work here illustrates the recent trend in SDN research of Moving from Simple Automation to Intelligent Systems that can Interpret User Intent, Reason Network Requirements, and Generate Corresponding Machine-Readable Network Configuration. While Agentic AI systems are becoming highly intelligent, there are a number of challenges awaiting their solutions in future research.

Above all, Agentic AI systems such as large language models need robust security, good governance and human verification for critical processes, in order to prevent mistakes or even malicious configurations. Large language models also have the drawback of a high computational effort and of potential inconsistencies in their reasoning in very dynamic network environments. Future research therefore needs to focus on scalable multi-agent systems, secure autonomous execution of actions on better explainability of decisions and on the integration into AI-native 6G-networking architectures.

Based on the provided literature, the Agentic AI frameworks are moving SDN from mere automation to a completely new generation of Network Management, where future SDN networks will become fully autonomous, highly scalable and above all user-centric, thanks to the use of Agentic AI to generate and deploy network configurations, using tools such as Natural Language Understanding, Multi-Agent Systems, Retrieval-Augmented Generation and Infrastructure-as-Code.

### 3. Research Gap

Recent advancements in Large Language Models (LLMs), Generative Artificial Intelligence (GenAI) and Intent-Based Networking (IBN) have further automated Software-Defined Networking (SDN). Current work is able to translate human written intents into machine-executable network configurations and even directly integrate these with current SDN controllers (e.g. OpenDaylight, ONOS, ETSI TeraFlowSDN). Emerging work is applying Retrieval-Augmented Generation (RAG), multi-agent systems and Infrastructure-as-Code (IaC) in order to improve contextual understanding and automate network deployment. As yet, however, a number of open research challenges need to be resolved in order to realize fully autonomous AI-based network management.

#### 3.1. Lack of End-to-End Autonomous Network

Configuration Most frameworks for network management using AI for configuring SDN include a human-in-the-loop, for example, approving a generated configuration for a network before it is applied. However, currently there is a lack of end-to-end

autonomous SDN configuration frameworks, i.e., systems that perform natural language interpretation and configuration generation but lack automated validation, deployment, continuous monitoring, and rollback of the generated configuration.

#### 3.2. Verifying System Constraints and Formal Proofs

Currently, in literature, assessing AI-generated network configurations, e.g. in terms of intent classification accuracy, execution time, and task completion rate, is common practice. Only a few studies check whether generated network configurations are correct before they are put into operation. Most of the important properties of network configurations, such as routing consistency, loop freedom, QoS (Quality of Service) compliance, policy consistency, and security constraints, are not checked using formal verification methods. Thus, the lack of powerful verification methods is a serious barrier to the deployment of AI-generated network configurations in production networks. Recently, the topic of verifiable configuration generation has received attention, but so far only for very specific scenarios.

#### 3.3. Vendor-Specific Implementation

Many current solutions are implemented on top of cloud platforms or on top of SDN controllers like ONOS, OpenDaylight, ETSI TeraFlowSDN or even on top of systems like Kubernetes. Hence, automatically generated configurations are very hard to port to environments with multi-vendor hardware like Cisco, Juniper, Huawei or even Arista equipment in mixed environments. Developing a vendor-agnostic AI framework that can generate standardized configurations for such environments is still a wide-open research challenge.

#### 3.4. Insufficient Domain Knowledge and Context Awareness

Most General-purpose LLMs suffer from a lack of knowledge about networking and, consequently, errors in the interpretation of network topologies, incorrect commands, and misconfigurations through hallucinations. Also, although RAGs have improved context awareness by incorporating domain-specific knowledge bases, their topology reasoning is, to date, not yet reliable enough for dynamic networks, for

changing routing policies, and for topology-aware reasoning in general. Recent studies also reveal that topology reasoning of existing systems is not yet reliable enough for generating network configurations that can be used in practice with confidence.

### 3.5. Security and Trustworthiness Issues

Use of Generative AI is increasing and with that, new security challenges are emerging, for example, prompt injection attacks, manipulation of malicious intent, generation of unauthorized configurations, and attacks with adversarial prompts. Most of the studies are improving the accuracy of automation, and rarely they discuss authentication, authorization, explainability, and secure policy management. Therefore, there is a need for security frameworks for secure and trustworthy AI-assisted SDN management, which has not been explored sufficiently so far.

### 3.6. Limited Explains of AI Decisions

Most AI-based SDN systems today are configured by these systems and the decisions with respect to routing and policy made by the AI-based system for the generated network configuration, lack of explanation of the decisions for network configuration by such systems, makes many network administrators not trust their AI-based SDN systems. Furthermore, the lack of explanations for the decisions of the configuration of the SDN-based networks makes troubleshooting of such networks very hard. Today, however, Explainable Artificial Intelligence (XAI) techniques are not yet given sufficient attention within the framework of AI-based SDN to be deployed in enterprises.

### 3.7. Future Research Direction

Future research should focus on the development of an intelligent, AI-driven SDN environment. This concept involves integrating existing technologies, including LLMs, multi-agent systems, Retrieval-Augmented Generation, formal verification, digital twins, and continuous learning, although the order of integration may vary. In the future, an SDN environment should enable vendor-independent configuration generation, autonomous deployment, real-time verification, adaptive optimization, and explainable decision-making. At the same time, strong security assurances are essential to

address unresolved challenges and ultimately achieve SDN environments that are fully autonomous while remaining trustworthy.

## 4. RESEARCH OBJECTIVES:

4.1 Identify and study current Network Configuration in SDN networks, and the problems with current practices of manual configuration as well as configuration based on rules.

4.2 To investigate how different AI methods and techniques could be used and designed to automatically and intelligently generate required SDN network configurations.

4.3 To design and develop a prototype system that is able to generate SDN configuration commands or policies by using AI.

4.4 Evaluate the AI-generated configuration in terms of accuracy and correctness. The network requirements and the capability of applying the configuration in a SDN-based environment shall be tested.

4.5 To compare between configurations generated by AI and human configurations in terms of time to configure, accuracy, consistency and chance of error in configuration.

4.6 To test the adaptability of the new method when the network requirements or even the network conditions change.

## 5. Research Methodology

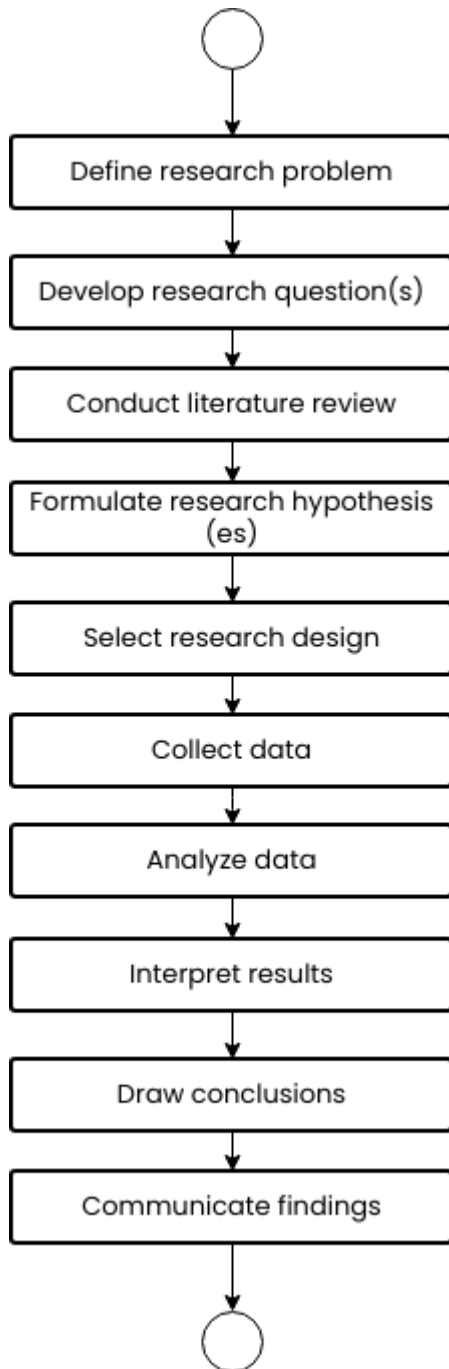


Fig -2: Referenced from Visual Paradigm

Table -1: Research Methodology Phases, Methods, and Expected Outputs

| Phase                  | Method                 | Output             |
|------------------------|------------------------|--------------------|
| Problem Identification | Gap Analysis           | Research problem   |
| Research Questions     | Question Formulation   | Research questions |
| Literature Review      | SLR                    | Research gaps      |
| Hypothesis             | Hypothesis Formulation | Hypotheses         |
| Research Design        | Design Selection       | Methodology        |
| Data Collection        | Data Collection        | Experimental data  |
| Data Analysis          | Statistical Analysis   | Results            |
| Interpretation         | Result Interpretation  | Key findings       |
| Conclusions            | Validation             | Conclusions        |
| Communication          | Dissemination          | Recommendations    |

## 6. CONCLUSIONS:

Software-Defined Networking (SDN) has significantly improved network management by separating the control plane from the data plane and providing centralized, programmable control of network resources. However, network configuration still requires considerable technical knowledge and manual effort, particularly when network requirements change frequently. This research examined the potential of Artificial Intelligence (AI), Generative AI, and Large Language Models (LLMs) to address these challenges by automatically generating network configurations from high-level, natural-language requirements. The literature review indicates that recent developments in LLM-based network management, Intent-Based Networking (IBN), Retrieval-Augmented Generation (RAG), multi-agent systems, and Infrastructure-as-Code (IaC) have created new possibilities for intelligent SDN configuration. Existing research demonstrates that AI can interpret user intent and generate machine-readable network configurations, reducing manual configuration effort and improving the efficiency of network management. However, the reviewed studies also identify important limitations, including configuration

errors and hallucinations, insufficient topology awareness, vendor-specific implementations, limited formal verification, security concerns, and the need for human intervention before configurations are deployed. Based on these findings, this research proposes an AI-driven approach for generating SDN network configurations from natural-language network requirements. The proposed approach aims to combine AI-based intent interpretation with configuration generation and validation so that generated configurations can be evaluated for accuracy, correctness, consistency, and applicability before deployment. The research also aims to compare AI-generated configurations with manually created configurations in terms of configuration time, accuracy, consistency, and error rate. The proposed research has the potential to reduce repetitive work for network administrators and provide a more accessible method of managing SDN environments. Nevertheless, AI-generated configurations should not be considered reliable for unrestricted autonomous deployment without appropriate validation and security mechanisms. Future development should therefore focus on formal verification, real-time network monitoring, vendor-independent configuration generation, explainable AI, secure AI execution, digital twins, and continuous feedback mechanisms.

In conclusion, AI-generated network configurations represent a promising direction for the future of SDN management. By combining SDN programmability with the natural-language understanding and configuration-generation capabilities of modern AI systems, network management can move from predominantly manual configuration toward more intelligent, automated, adaptive, and user-centric network operations. The successful implementation of such systems will depend not only on the accuracy of AI-generated configurations but also on their verification, security, explainability, and ability to operate reliably in dynamic and multi-vendor network environments.

## REFERENCES

### 1. SDN foundation

D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

### 2. IBN survey

A. Leivadreas and M. Falkner, "A Survey on Intent-Based Networking," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 625–655, 2023, doi: 10.1109/COMST.2022.3215919.

### 3. Intent-driven autonomous networking

K. Mehmood, K. Kravetska, and D. Palma, "Intent-driven autonomous network and service management in future cellular networks: A structured literature review," *Computer Networks*, vol. 220, p. 109477, 2023.

### 4. LLM-based intent configuration

N. Tu, S. Nam, and J.-K. Hong, "Intent-Based Network Configuration Using Large Language Models," *International Journal of Network Management*, vol. 35, no. 1, p. e2313, 2025, doi: 10.1002/nem.2313.

### 5. Agentic AI + IBN + IaC

D. Brodimas, A. N. Birbas, D. Kaposos, and S. Denazis, "Intent-Based Infrastructure and Service Orchestration Using Agentic-AI," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7150–7168, 2025, doi: 10.1109/OJCOMS.2025.3600706.

### 6. Generative AI + IBN

D. Adanza, L. Gifre, P. Alemany, C. Natalino, P. Monti, R. Muñoz, and R. Vilalta, "Leveraging generative AI for intent-based networking operations in network slices," *Computer Networks*, vol. 272, p. 111647, 2025, doi: 10.1016/j.comnet.2025.111647.

### 7. Verifiable LLM configuration

C. Lin, L. Zhu, H. Liu, D. Zhang, J. Yu, and L. Liao, "DeepConfig: A verifiable configuration generation framework for MAN overlays using LLMs," *Computer Networks*, vol. 280, p. 112177, 2026, doi: 10.1016/j.comnet.2026.112177.

### 8. LLM network-management survey

F. Liu, B. Farkiani, and P. Crowley, "Large Language Models for computer networking operations and management: A survey on applications, key techniques, and opportunities," *Computer Networks*, vol. 271, p. 111614, 2025, doi: 10.1016/j.comnet.2025.111614.

### 9. LLM-based network management survey

J. Hong, N. V. Tu, and J.-W. K. Hong, "A Comprehensive Survey on LLM-Based Network Management and Operations," *International Journal of Network Management*, vol. 35, no. 6, p. e70029, 2025, doi: 10.1002/nem.70029.

#### 10. NetConfEval

C. Wang, M. Scazzariello, A. Farshin, S. Ferlin, D. Kostić, and M. Chiesa, "NetConfEval: Can LLMs Facilitate Network Configuration?," *Proceedings of the ACM on Networking*, vol. 2, CoNEXT2, Art. no. 7, 2024, doi: 10.1145/3656296.

#### 11. NETCONF/YANG configuration

"Generative AI for Low-Level NETCONF Configuration in Network Management Based on YANG Models," in *Proc. 20th IEEE Int. Conf. Network and Service Management (CNSM)*, 2024, doi: 10.23919/CNSM62983.2024.10814410.

#### 12. Automated configuration generation

C. Liu, X. Xie, X. Zhang, and Y. Cui, "Large Language Models for Networking: Workflow, Advances and Challenges," *IEEE Network*, 2024, doi: 10.1109/MNET.2024.3510936.

#### 13. Natural-language network management

S. K. Mani *et al.*, "Enhancing Network Management Using Code Generated by Large Language Models," in *Proc. 22nd ACM Workshop on Hot Topics in Networks (HotNets)*, 2023, pp. 196–204, doi: 10.1145/3626111.3628183.

#### 14. Router configuration correctness

R. Mondal, A. Tang, R. Beckett, T. Millstein, and G. Varghese, "What do LLMs need to Synthesize Correct Router Configurations?," in *Proc. 22nd ACM Workshop on Hot Topics in Networks (HotNets)*, 2023, doi: 10.1145/3626111.3628194.

#### 15. Formal verification of SDN

A. Souri, M. Norouzi, P. Asghari, A. M. Rahmani, and G. Emadi, "A systematic literature review on formal verification of software-defined networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3788, 2020, doi: 10.1002/ett.3788.

#### 16. IBN conflict detection

X. Zheng, A. Leivadeas, and M. Falkner, "Intent Based Networking management with conflict detection and policy resolution in an enterprise network," *Computer Networks*, vol. 219, p. 109457, 2022, doi: 10.1016/j.comnet.2022.109457.

#### 17. Zero-touch IBN

B. Martini, M. Gharbaoui, and P. Castoldi, "Intent-based zero-touch service chaining layer for software-defined edge cloud networks," *Computer Networks*, vol. 212, p. 109034, 2022, doi: 10.1016/j.comnet.2022.109034.

#### 18. Intent-based networking survey

E. Zeydan and Y. Turk, "Recent Advances in Intent-Based Networking: A Survey," in *Proc. IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5, doi: 10.1109/VTC2020-Spring48590.2020.9128422.

#### 19. Security-intent extraction

R. H. Ribeiro *et al.*, "A deterministic approach for extracting network security intents," *Computer Networks*, vol. 214, p. 109109, 2022, doi: 10.1016/j.comnet.2022.109109.

#### 20. LLM configuration translation

Y. Wei, X. Xie, Y. Zuo, T. Hu, X. Chen, K. Chi, and Y. Cui, "Leveraging LLM Agents for Translating Network Configurations," arXiv:2501.08760, 2025, doi: 10.48550/arXiv.2501.08760.

#### 21. LLMs + next-generation networking

C.-N. Hang, P.-D. Yu, R. Morabito, and C.-W. Tan, "Large Language Models Meet Next-Generation Networking Technologies: A Review," *Future Internet*, vol. 16, no. 10, p. 365, 2024, doi: 10.3390/fi16100365.

## BIOGRAPHIES



Mr. Tushar Mohan Sawant

M.Sc. Information Technology  
IT/CS Dept. D G Ruparel College