2024 IEEE/IFIP Network Operations and Management Symposium (NOMS 2024) - Poster

ChatGPT-enabled Network Automation using API-based Prompts

Olasupo Okunaiya
College of Computing
Birmingham City University
Birmingham, United Kingdom
olasupo.o@gmail.com

Ron Austin
College of Computing
Birmingham City University
Birmingham, United Kingdom
ron.austin@bcu.ac.uk

Shao Ying Zhu
College of Computing
Birmingham City University
Birmingham, United Kingdom
pzhu5500@gmail.com

***Abstract*—In the rapidly evolving world of network provisioning and management, the traditional approach to device configuration has proven to be error-prone and time-consuming, especially in large-scale deployments. The emergence of network automation has sought to address these challenges by leveraging software to perform tasks which were traditionally executed manually by network engineers. However, utilizing these tools requires technical expertise. This study explores the potential of using Large Language Models (LLMs), specifically ChatGPT, to generate Ansible Playbooks for automating network deployment tasks. The goal is to eliminate the need for network engineers to prepare automation scripts. An application named "Bubbln" was developed to prompt ChatGPT through APIs and execute the ChatGPT-generated playbooks on a network of four Cisco routers. A systematic evaluation of this approach reveals impressive results including 99% accuracy, 100% reliability, a 62% improvement in efficiency compared to traditional approach and a 13.52% efficiency gain over manually prepared playbook approach. These results highlight ChatGPT as a promising tool to enhance existing network automation approaches.**

Keywords—ChatGPT, chatbot, network automation, provisioning, configuration.

I. INTRODUCTION

The evolution of network provisioning processes has transitioned from a traditional, error-prone approach to the use of automation tools. Initially, engineers configured each networking device manually through Command Line Interfaces, which became impractical for large-scale networks. The growing complexity of networks necessitated faster setup procedures, leading to the concept of network automation. Network automation involves the utilization of software tools such as Ansible, Chef, Puppet, and others to streamline tasks that were traditionally performed manually by network engineers [1].

Recent research on network automation approaches has primarily focused on the preparation of Ansible [1]-[7] and Python scripts [8]-[20] by the researchers as summarized in Table I, to execute various network tasks. For instance, [2] compared the performance of Ansible, Salt and Puppet, by preparing scripts for different tools, [3] utilized Ansible scripts to configure both the interfaces and routing protocols on a network of five Cisco and five MikroTik routers, while [4] utilized Ansible scripts to provision and verify routing protocols on a network of three routers. In a similar manner, Python scripts were prepared by [8] to prove its ability to verify network states automatically and [9] utilized python in a proposed architecture to automate VPN configurations. Although this approach of manually preparing automation scripts offer numerous advantages over traditional methods, including increased efficiency and a reduction in errors, there is a requirement for technical expertise, for effective use of the automation tools [2]. Engineers need to prepare or tailor automation scripts to meet network requirements, leading to potential for syntax errors [3].

TABLE I. RELATED WORKS ON NETWORK AUTOMATION

Reference	Python	Ansible	ChatGPT
[1]		✓	×
[2]		✓	×
[3]		✓	×
[4]		✓	×
[5]		✓	×
[6]		✓	×
[7]		✓	×
[8]	✓		×
[9]	✓		×
[10]	✓		×
[11]	✓		×
[12]	✓		×
[13]	✓		×
[14]	✓		×
[15]	✓		×
[16]	✓		×
[17]	✓		×
[18]	✓		×
[19]	✓		×
[20]	✓		×

More recently, significant advancements have emerged in the field of Artificial Intelligence, with a particular focus on Large Language Models (LLMs) such as ChatGPT and Bard. These models exhibit the remarkable ability to process instructions expressed in natural language to generate desired textual outputs [21]. The release of ChatGPT has ushered in a new era of LLM applications across various domains, including software coding [22], debugging [23], vulnerability detection [24] and more. Considering these advancements, research on the use of LLMs to automate the scripting process did not return any result. This prompted our research to explore the potential of leveraging ChatGPT to generate Ansible playbooks for network deployment, eliminating the necessity for manual preparation of automation scripts. This approach is expected to improve network automation efficiency and ultimately increase the productivity of engineers.

II. METHODOLOGY

To assess the potential of ChatGPT to generate Ansible playbooks, a systematic methodology was devised. The key steps of the methodology included the following:

A. Obtain API Keys

The first step was to register on the OpenAI portal to obtain an API key, which was then utilized in the application developed for prompting ChatGPT.

B. Application Development

An application named Bubbln, was developed using Python programming language to get network parameters from users, compose prompts, prompt ChatGPT, process the response from ChatGPT, and finally execute the ChatGPT-generated playbooks on a network of four Cisco routers. Bubbln was developed with three main components as follows:

- Logic and Control – This component manages taking and validating router parameters from users. It then combines these inputs with a pre-defined set of instructions written in natural language to form the prompts illustrated in Fig. 2, which were then sent to the OpenAPI Gateway. It also receives ChatGPT's raw response from the OpenAPI gateway component and extracts the playbooks, which are subsequently sent along with the host file to the Playbook Executor component for onward execution on the network. This entire process is illustrated in Bubbln's architecture in Fig. 1.

- OpenAPI Gateway – This component is responsible for prompting and receiving responses from ChatGPT through APIs. It utilizes GPT-4 model, with a temperature parameter setting of zero and maximum token of 1500 and OpenAI's chatCompletion API for sending prompts to ChatGPT. The choice of parameter values was to limit the tendency of ChatGPT generating unwanted features. It works by passing prompts received from the Logic and Control component to ChatGPT and sends the raw response from ChatGPT, which contains a combination of the playbooks and explanation notes, back to the Logic and Control component for processing as illustrated in Fig. 1.

- Playbook Executor – This component handles the execution of the ChatGPT-generated playbooks on the network. It makes use of the Python library called Ansible-runner which takes a hosts file and the ChatGPT-generated playbooks as input from the Logic and Control component.

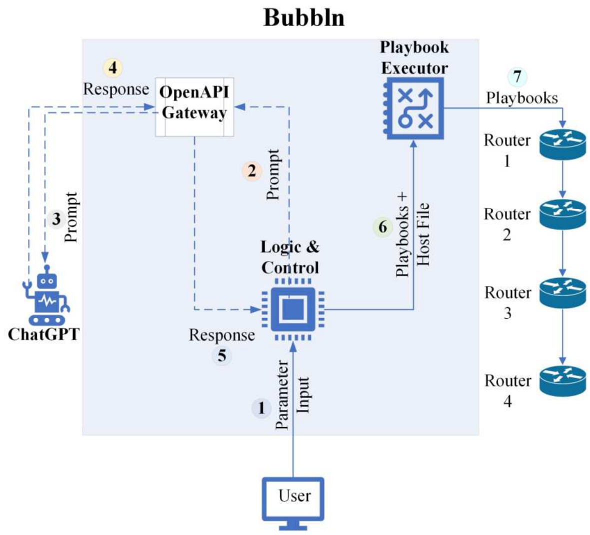

Fig. 1. Bubbln's architecture and process flow.

Prompt for Router 4

Requirements: Strictly adhere to the following explicitly stated requirements; Write a simple Ansible playbook with separate tasks for each protocol and interface configurations with the following details; hosts: R4 Do not worry about the inventory file; Ensure each Task is named; Ensure all generated playbooks adhere to yaml's rule of always starting a playbook with \`---\` and ending the playbook with a new line containing \`...\`; Always use ios_config module and ensure unsupported parameters are not generated; Use \`parents\` argument to implement stanzas; when configuring interfaces, ensure you generate codes for only provided interfaces and always implement 'No Shutdown' for each interface; when configuring routing protocols, ensure you generate codes for only provided protocols and that the protocol is initialized only under the parents argument using the format \`router protocol-type xx\`; set \`replace\` argument to block. \`replace\` argument should always be child to \`ios_config\`; Protocol: ospf OSPF Area: 20, Process ID: 1, Number of networks to advertise: 3 network1: 192.168.6.2 0.0.0.3 network2: 192.168.70.1 0.0.0.255 network3: 192.168.80.1 0.0.0.255 Interface: lo0, IP: 192.168.70.1 255.255.255.0 Interface: lo1, IP: 192.168.80.1 255.255.255.0

Derived from a combination of iteration value and "R" during prompt generation for each router

Fixed Set of instructions to guide ChatGPT on playbook generation

Dynamic parameters derived from user input

Fig. 2. Prompt for generation of playbook for router four.

C. Network Design

A network of four Cisco routers with parameter details displayed in Fig. 3, was set up to test the ChatGPT-generated playbooks. It was also used to conduct traditional and user-prepared playbook experiments. The physical interfaces, shown in the topology, static routes, SSH2.0 and Ansible host file, were manually configured for the ChatGPT-generated and user-prepared playbook experiments to enable reachability from the controller machine, which hosts Bubbln and Ansible installations.

D. Conduct of the Experiment

Three categories of experiments were conducted. The main experiment, which is ChatGPT-enabled, and two other benchmark experiments as follows:

- ChatGPT-generated Playbooks: Twenty-five experiments were conducted to test the defined performance metrics. It involved taking parameter input from users, prompting ChatGPT for playbook generation, automatically executing the generated playbooks on the network and keeping a record of the deployment timings. Inputting of parameters was done only in the first experiment and the loaded parameters were re-used for the remaining twenty-four experiments.

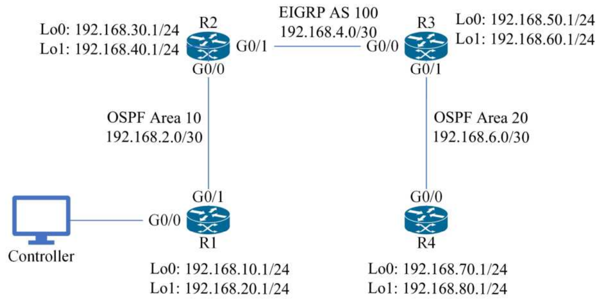

Fig. 3. Network topology and parameters used for the experiments.

- User-prepared Playbooks: A single experiment was conducted, where Ansible scripts were manually updated with the required network parameters. A measurement of the time taken to edit the scripts as well as execution time were recorded.
- Traditional Manual Configuration: A final experiment was conducted to manually configure the network. The deployment duration was also recorded.

E. Defining Performance Metrics

Evaluating the potential of the ChatGPT-enabled approach relies on performance metrics. In this study, three key indicators were utilized to assess the outcome of the experiment:

- Accuracy: This measures the correctness of the ChatGPT-generated playbooks and was assessed by a successful execution of the playbooks without runtime errors.
- Efficiency: This quantifies the average time taken to deploy the network compared to traditional and user-prepared playbook approaches. Equation (1) was used to determine the calculation for efficiency.
- Reliability – This evaluates the fulfilment of the intended purpose for each playbook. It ensured parameters and features in the generated playbooks aligned with those provided in the prompt.

F. Time Tracking

The time taken for inputting configuration parameters, generating playbooks, and executing playbooks were tracked in Bubbln using the time python library. These served as input towards evaluating efficiency of this approach.

III. The Result

This section offers a detailed account of the outcomes derived from the experiments conducted as outlined in Table II.

A. Reliability Assessment

Each ChatGPT-generated playbook with zero runtime error fulfilled its intended purpose, thereby resulting in a reliability outcome of 100%. A sample of the generated playbook can be seen in Fig. 4.

B. Accuracy Evaluation

During the execution of the twelfth experiment, a runtime failure was encountered due to a syntax error in the

```
---
- hosts: R4
  gather_facts: no
  tasks:
    - name: Configure interface lo0
      ios_config:
        lines:
          - ip address 192.168.70.1 255.255.255.0
          - no shutdown
        parents: interface lo0
        replace: block

    - name: Configure interface lo1
      ios_config:
        lines:
          - ip address 192.168.80.1 255.255.255.0
          - no shutdown
        parents: interface lo1
        replace: block

    - name: Configure OSPF
      ios_config:
        lines:
          - network 192.168.6.2 0.0.0.3 area 20
          - network 192.168.70.1 0.0.0.255 area 20
          - network 192.168.80.1 0.0.0.255 area 20
        parents: router ospf 1
        replace: block
...
```

Fig. 4. ChatGPT-generated playbook for router four.

TABLE II. Detailed Outcome Of Each Experiment

Experiment	Parameter Input Time (Seconds)	Playbook Generation Time (Seconds)	Execution Time (Seconds)	Deployment Time (Seconds)
1	238.34	172.63	61.19	472.16
2	238.34	171.07	62.78	472.19
3	238.34	161.13	64.44	463.91
4	238.34	151.27	63.63	453.24
5	238.34	227.48	62.11	527.93
6	238.34	215.3	61.8	515.44
7	238.34	148.1	63.55	449.99
8	238.34	175.65	62.85	476.84
9	238.34	183.06	62.1	483.5
10	238.34	191.62	61.82	491.78
11	238.34	186.62	62.65	487.61
12	238.34	146.15	62.08	446.57
13	238.34	158.28	63.25	459.87
14	238.34	166.7	61.11	466.15
15	238.34	167.13	61.49	466.96
16	238.34	161	62.48	461.82
17	238.34	123.67	62.19	424.2
18	238.34	173.24	61.55	473.13
19	238.34	143.92	61.21	443.47
20	238.34	175.68	61.95	475.97
21	238.34	111.73	61.74	411.81
22	238.34	144.39	62.34	445.07
23	238.34	183.6	63.72	485.66
24	238.34	169.47	62.25	470.06
25	238.34	177.27	64.78	480.39
AVERAGE (ChatGPT-generated)	238.34	167.45	62.44	468.23
User-prepared Approach	479.54	-	62	541.54
Traditional Approach	-	-	-	1219

generation of router two's playbook. As a result, the overall accuracy metric dropped to 99%.

C. Efficiency Analysis

The ChatGPT-enabled approach took an average time of 468.23 seconds, encompassing parameter input, playbook generation, and execution across twenty-five experiments. In contrast, the user-prepared playbook experiment required 541.54 seconds, involving template editing and playbook execution, while the deployment time for the traditional approach was 1219 seconds. The details can be seen in Fig. 5.

Based on the deployment timings, an efficiency calculation was performed with the formula:

$$\text{Efficiency} = \left(1 - \frac{\text{Time Taken by Approach A}}{\text{Time Taken by Approach B}}\right) * 100 \tag{1}$$

The ChatGPT-enabled approach is 61.63% more efficient than the traditional approach and 13.52% more efficient than user-prepared playbook approach.

IV. Evaluation

Traditional network deployment approach is error-prone and inefficient, prompting the adoption of automation tools, which also require technical expertise to operate and prone to syntax errors. This led to our investigation into ChatGPT's potential to generate Ansible playbooks. This section discusses the discovered pros and cons of using a ChatGPT-enabled tool for automating network configuration tasks.

A. Pros of A Chat-GPT Network Automation Tool

The outcome of this research highlights a ChatGPT-enabled approach as a potential tool for enhancing the existing network automation tools for the following reasons:

- Simplified Automation Process: The ChatGPT-enabled approach seamlessly generated Ansible playbooks with 99% accuracy and 100% reliability as seen in Fig. 6. This result positions the approach as promising for eliminating manual scripting, thereby reducing time spent on preparing Ansible playbooks. It addresses the limitations identified by [2] and [3] and obviates the need for extensive technical expertise, eliminating limitations encountered by [2].
- Efficiency: The ChatGPT-enabled approach exhibited notable efficiency gains. On average, it took 468.23 seconds, outperforming the user-prepared playbook approach (541.54 seconds) by 13.52% and the traditional approach (1219 seconds) by 61.63%. It is noteworthy that enhancing the efficiency of a ChatGPT-enabled approach could be achieved through the adoption of a less time-intensive parameter input method, such as the direct loading of parameters from design documents.

B. Cons of A Chat-GPT Driven Network Automation Tool

During this experiment, certain issues were encountered which need to be considered before adopting this approach to automate a network. These considerations include the following:

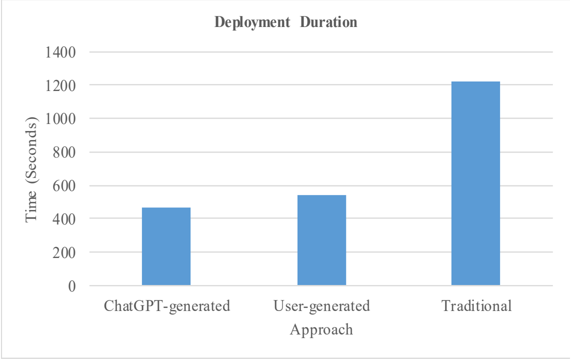

Fig. 5. Deployment time for each approach.

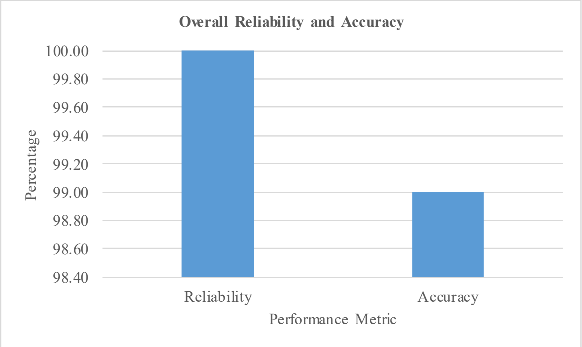

Fig. 6. Experimental outcome of reliability and accuracy of ChatGPT-generated playbooks.

- Cost: The utilization of ChatGPT APIs involves incurring costs based on usage volume. In essence, a higher frequency of prompts sent to ChatGPT results in proportionally increased costs. This financial aspect must be carefully factored into the decision to employ this approach.
- Rate Limit: The process of transmitting prompts to ChatGPT through APIs is subject to predefined rate limits, necessitating thoughtful consideration during the design of an automation tool. This is vital to prevent the unforeseen interruption of automation processes.

V. Conclusion And Future Work

In conclusion, the experiments undertaken to assess the potential of employing ChatGPT for generating playbooks for network deployment tasks have yielded promising and noteworthy results. ChatGPT has exhibited a commendable accuracy rate of 99% and an exceptional reliability rate of 100%. Moreover, these findings indicate an appreciable improvement in efficiency, with ChatGPT streamlining processes by approximately 62% when compared to traditional methods and 14% when compared to user-prepared playbook approaches. The combination of these results demonstrates ChatGPT as a potential tool to enhance existing network automation approaches.

In future research, we aim to investigate ChatGPT's potential for network fault monitoring and auto-recovery to simplify network management, enhance resilience, and minimize downtime. This marks a significant advancement in applying AI and natural language processing to network automation, opening avenues for further exploration of ChatGPT's potential in network management.

REFERENCES

[1] J. Wijaya, "Network Automation Using Ansible for Cisco Router Basic Configuration," Retrieved from osf. io/u8cdm, 2018.

[2] S. Wågbrant and V. Dahlén Radic, "Automated Network Configuration: A Comparison Between Ansible, Puppet, and SaltStack for Network Configuration," Mälardalen University Digital Archive, 2022. [Online]. Available: urn:nbn:se:mdh:diva-58886. (OAI: oai:DiVA.org:mdh-58886)

[3] M. F. Islami, P. Musa, and M. Lamsani, "Implementation of Network Automation using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI: Array," Jurnal Ilmiah KOMPUTASI, vol. 19, no. 2, pp. 127-134, 2020

[4] M. F. Mohd Fuzi, K. Abdullah, I. H. Abd Halim, and R. Ruslan, "Network automation using ansible for EIGRP network," Journal of Computing Research and Innovation (JCRINN), vol. 6, no. 4, pp. 59-69, 2021.

[5] I. Ortiz-Garcés, A. Echeverría, and R. Andrade, "Automation Tasks Model for Improving Hardening Levels on Campus Networks," in 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), July 2021, pp. 30-35, IEEE.

[6] E. Mulyana and G. Fakih, "Network Automation with a Single Source of Truth in a Heterogeneous Environment," International Journal on Electrical Engineering and Informatics, vol. 14, no. 1, 2022.

[7] A. Khumaidi, "Implementation of Devops Method for Automation of Server Management Using Ansible," Jurnal Transformatika, vol. 18, no. 2, pp. 199-209, 2021.

[8] A. Elezi and D. A. Karras, "On Detailed Network Systems Configuration Management Automation using Python," WSEAS TRANSACTIONS ON COMMUNICATIONS, 2023.

[9] L. Firdaouss, B. Ayoub, M. Manal, and Y. Ikrame, "Automated VPN configuration using DevOps," Procedia Computer Science, vol. 198, pp. 632-637, 2022.

[10] O. W. J. Altalebi and A. A. Ibrahim, "Optimization of Elapsed Time of Automation for Large-Scale Traditional Networks and Proposing New Automation Scripts," in 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 2022, pp. 1-10, doi: 10.1109/HORA55278.2022.9799873.

[11] M. Al-Mekhlal, A. AlYahya, A. Aldhamin, and A. Khan, "Network Automation Python-based Application: The performance of a Multi-Layer Cloud Based Solution," in 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS), August 2022, pp. 1-8, IEEE.

[12] A. L. Ptr and B. H. Hayadi, "IOT Network Automation Implementation in Router Configuration Handling: Implementasi IOT Network Automation Pada Penanganan Konfigurasi Router," Jurnal MESIL (Mesin Elektro Sipil), vol. 3, no. 1, pp. 13-21, 2022.

[13] A. M. Mazin, R. Ab Rahman, and M. Kassim, "Performance analysis on network automation interaction with network devices using python," in 2021 IEEE 11th IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), April 2021, pp. 360-366, IEEE.

[14] P. Mihăilă, T. Bălan, R. Curpen, and F. Sandu, "Network Automation and Abstraction using Python Programming Methods," MACRo 2015, vol. 2, no. 1, pp. 95-103, 2017.

[15] L. G. Mauboy and T. Wellem, "Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik," JURIKOM (Jurnal Riset Komputer), vol. 9, no. 4, pp. 790-799, 2022.

[16] K. Nugroho, A. D. Abrariansyah, and S. Ikhwan, "Performance Comparison between Paramiko and Netmiko Libraries in Network Automation Process," InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, vol. 5, no. 1, pp. 1-8, 2020.

[17] I. Ortiz-Garcés, A. Echeverría, and R. Andrade, "Automation Tasks Model for Improving Hardening Levels on Campus Networks," in 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), July 2021, pp. 30-35, IEEE.

[18] L. G. Mauboy and T. Wellem, "Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik," JURIKOM (Jurnal Riset Komputer), vol. 9, no. 4, pp. 790-799, 2022.

[19] M. Shyam and S. Dr.Ambareesh, "NETWORK AUTOMATION," International Journal of Innovative Research in Advanced Engineering, vol. 9, pp. 203-206, 2022. DOI: 10.26562/ijirae.2022.v0908.10.

[20] A. F. Rochim, A. Rafi, A. Fauzi, and K. T. Martono, "As-RaD System as a Design Model of the Network Automation Configuration System Based on the REST-API and Django Framework," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, pp. 291-298, 2020.

[21] O. Suissa, A. Elmalech, and M. Zhitomirsky-Geffet, "Text analysis using deep neural networks in digital humanities and information science," Journal of the Association for Information Science and Technology, vol. 73, no. 1, 2021, doi: 10.1002/asi.24544.

[22] P. Abrahamsson et al., "ChatGPT as a Fullstack Web Developer - Early Results," Tampere University, Finland, 2023.

[23] H. Tian et al., "Is ChatGPT the Ultimate Programming Assistant—How far is it?" arXiv preprint arXiv:2304.11938 [cs.SE], 2023. Available at: https://doi.org/10.48550/arXiv.2304.11938.

[24] M. Omar and S. Shiaeles, "VulDetect: A novel technique for detecting software vulnerabilities using Language Models," 2023 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 2023, pp. 105-110, doi: 10.1109/CSR57506.2023.10224924.