JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 2 | December 2023: 222-231
DOI: 10.15575/join.v8i2.1157

Automate IGP And EGP Routing Protocol Configuration Using A Network Automation Library

Yuansa Alfaresa[1], Bongga Arifwidodo[2], Fauza Khair[3]
[1,2,3]Department of Telecommunication Engineering, Institut Teknologi Telkom Purwokerto, Indonesia

Article Info

Article history:

Received August 04, 2023
Revised October 22, 2023
Accepted October 23, 2023
Published December 28, 2023

Keywords:

BGP
Network Automation
OSPF
Paramiko
Telnetlib

ABSTRACT

Data communication is sending data from client to client through a computer network. The increasing use of data communication makes computer networks more complex. Complex computer networks make it difficult for network administrators to configure them, especially routing protocol configuration. Network administrators are in charge of configuring routing protocols and managing networks. In addition, the more devices on the network, the greater the chance of human error from the administrator. Therefore, network automation is one solution that helps network administrators overcome this. This study focuses on analyzing the performance of network automation using the Paramiko and Telnetlib libraries. The routing protocol used by OSPF for IGP and BGP for EGP. The scenario in this study involves configuring IP addresses and configuring OSPF and BGP routing. Based on the test results, the Telnetlib library is better than the Paramiko library in terms of script delivery time, convergence time, and delay by 19.237% when applied to the IGP and EGP routing protocols.

Corresponding Author:

Bongga Arifwidodo
Department of Telecommunication Engineering, Institut Teknologi Telkom Purwokerto
Jl. DI Panjaitan No.128, Banyumas, Indonesia 53147
Email: bongga@ittelkom-pwt.ac.id

1. INTRODUCTION

Data communication is the process of sending data through a computer network. Computer networks today are increasingly complex and complicated. Several network devices, including routers of various types and brands, are indispensable for building a network. The increasing complexity of a network makes it difficult for network administrators to configure, plus the many types, types, and brands of routers configured. With this difficulty, it can make network administrators make mistakes in configuration or human error. In addition to human error, another problem that appears in network administrators is the length of time to configure. More effective methods for configuring and managing networks are indispensable to network administrators [1]-[2]. The complexity of a network takes longer if it has to be manually configured one by one on many routers used in large network areas [3]. So that makes the computer network system needs to be managed properly[4].

One way that can help network administrators in configuring is Network Automation. The process of automating the setup (configuration), management, testing, and use of network devices, including physical and virtual, is known as Network Automation [5]. The use of Network Automation generally uses a script written in a programming language [6]. Network and system administrators often use scripting languages to automate their configuration. Then time, energy, and human error will be reduced. Python and Ansible are programming languages often used in network automation processes[7]-[8]. Python is one of the most extensible high-level programming languages. Network Automation works to determine effective ways to map, configure, and manage networks. Network Automation uses an API (Application Programming Interface), which connects one application with

another. This API-based automation replaces CLI (Command Line) commands where calls from APIs are made using programming languages such as Python and Ansible[9]. Python has many packages, modules, and libraries that have been written to provide conveniences such as development and network automation so that it can make it easier for users[10]-[11].

Several libraries in the Python programming language are used in network automation, such as the Paramiko and Telnetlib libraries. The Paramiko library is an implementation of the SSHv2 protocol in Python. SSHv2 consists of a server and a client[12]. Paramiko offers encryption methods when making remote connections and file transfers, thus making data communication between clients more secure[13]. Paramiko library supports several network vendors or network devices, such as Arista, Cisco, and Juniper[14]. The telnetlib library is a standard module of the Python programming language. This library implements remote functionality with the telnet protocol[15]. Telnetlib can be used when a remote network uses the telnet protocol; therefore, this library is suitable for use in embedded systems[16].

A review of previous research was conducted to provide an overview of this research. The study [5]compares network automation performance in Paramiko and Netmiko libraries. This test was conducted to determine the average value of script execution time from four scenarios in the form of Static Routing, Dynamic Routing in the form of RIPv2, Firewall in the form of NAT (Network Address Translation), and SNMP (Simple Network Management Protocol). The scenario in the study was carried out on two different topologies, the first topology using two Mikrotik and the second using three Mikrotik. Furthermore, research [17] compares network automation performance in both libraries, namely Paramiko and Netmiko libraries. Tests are conducted to determine the length of script delivery time, network convergence time, and quality of service (QoS) data. The test was carried out using a ring topology composed of four routers. The configuration sent to find out the script delivery time and convergence time is the OSPF routing protocol configuration. The test results found that the paramiko library is 4.14 times faster than the netmiko library in script delivery time and convergence parameters. Then research [18] testing Python libraries for network automation, namely Telnetlib, Paramiko, and Netmiko libraries. This study aims to determine the advantages and disadvantages of each library. The study states that the Telnetlib library is easier to use.

There is [19] research discusses using Paramiko libraries for network automation and the Django framework for web interfaces. The results of this study are applications that can be used to automate networks in terms of configuring static routing and dynamic routing, creating VLANs, and carrying out maintenance in the form of backups and restores carried out centrally. Research [6] using the Paramiko library for web-based network automation. The goal is to create a web-based dashboard that can only manage routers through one interface. The results of this research are a web-based network automation application using the Python programming language with the Django framework and the Paramiko library. The application was tested on six routers: three Cisco routers and three Mikrotik routers.

This study aims to determine the performance of network automation in Python libraries, namely Paramiko and Telnetlib, which are applied to IGP and EGP routing protocols. In this study, the Paramiko and telnetlib libraries were chosen because both libraries only use one command execution stage and have the same structure. Unlike the previous study, which only focused on one routing protocol, this study used two routing protocols: Internal Gateway Protocol (IGP) in the form of OSPF and Exterior Gateway Protocol (EGP) in the form of BGP. To find out the performance of the two libraries seen from the data to be retrieved, namely script delivery time, convergence time, delay, and throughput. The use of two routing protocols is also to find out whether the library has the same performance on each routing protocol.

2. METHOD

The author researched Network Automation Performance Analysis on Paramiko and Telnetlib Libraries for IGP and EGP Routing Protocols in this study. This research uses GNS3 Software to perform simulations, VMWare Workstation as GNS3 VM to be used as a server by Network Automation ubuntu (Ubuntu Server) on GNS3, and Wireshark, which is used to perform data retrieval.

Figure 1 shows the flow diagram of the system design in this study. The first step in the research is to conduct a literature study of several studies related to Network automation for routing protocol configuration and other materials related to this research by comparing several related journals and making comparisons to determine the title and the focus of this study. In addition to comparing and determining the focus or title of the study, this stage also serves to understand the basic concepts of the topic. Next is to design a topology with a Full Mesh Topology compiled using GNS3 simulator software.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Before compiling the topology, researchers installed a Docker Network automation system that already supports network automation. The topology comprises eleven routers, one switch, two clients, and an Ubuntu Network automation system.

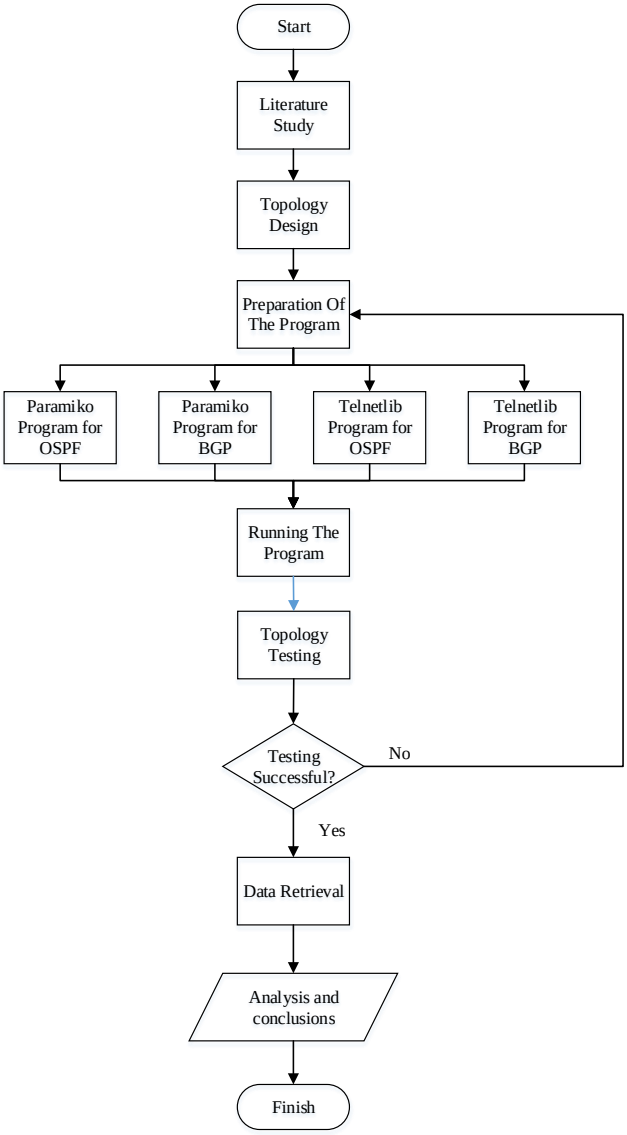

Figure 1. Research Flowchart

After compiling the topology, the next step is compiling a script or program line using Python. Four scripts will be used: the Paramiko library for IGP and EGP routing and the Telnellib library for IGP and EGP routing. Python programs are created in the Ubuntu Network Automation system by creating files with the "nano" command to enter the files that have been created. After the program is created, the next step is to run each of the four programs that have been created, whether there is an error or not. Running the program is done on the Ubuntu Network Automation system with the command "python3". If there is an error, then it is necessary to return to the program preparation process to fix errors in the program that has been created. If there is no error, it means the test was successful.

When the program test is successfully run without any errors, the next step is data retrieval. The data to be retrieved is in the time needed for Network Automation Ubuntu to provide configuration to the router, IGP and EGP convergence time, and Quality of Service data retrieval in the form of Delay and Throughput. The process of retrieving the three data sets is carried out using the Wireshark software. After all the data is collected, proceed with the analysis steps. The analysis was carried out by comparing

data obtained from the Paramiko library and Telnetlib in graphs, and conclusions were drawn when the analysis process was completed.

2.1 Topology Design

This research uses network topology to automate IGP and EGP routing protocols. The topology can be seen in Figures 1 and 2, where the topology used is a Full Mesh topology. The topology comprises eleven Cisco 7200 routers, one switch, two client PCs, and one Ubuntu Network Automation system. Ubuntu's Network Automation system serves as a place to create Python programming scripts for the Paramiko and Telnetlib libraries. The system is also used to transmit the script to all routers through the intermediary of the switch device. In addition to connecting the network automation system with the router, the switch also functions in retrieving data to find out the time needed to provide configuration.

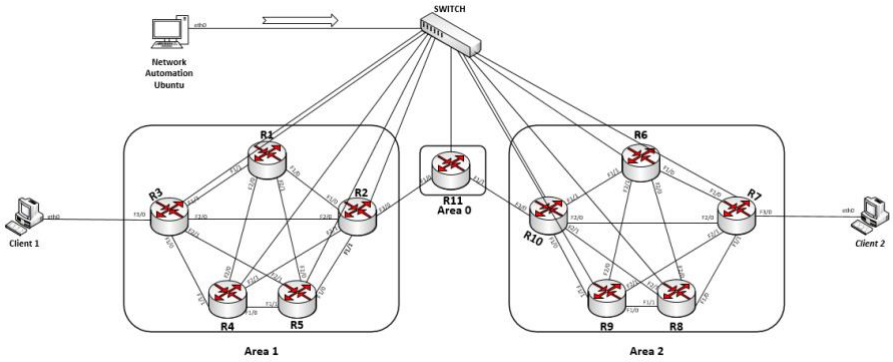

Figure 2. IGP Network Topology

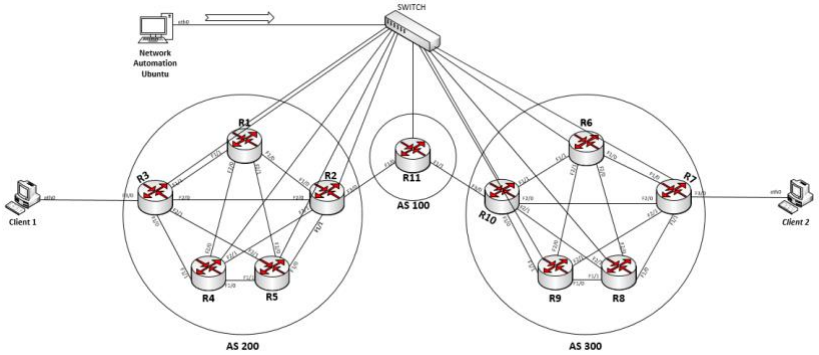

Figure 3. EGP Network Topology

Figure 1 is the topology that will be used for testing Paramiko and Telnetlib libraries in IGP routing, namely the OSPF routing protocol. Using OSPF, the type of area used is multi-area, with three areas: area 0 (Backbone), area 1, and area 2. Nodes R11 to R2 and R10 are in area 0 (Backbone). R1, R2, R3, R4 and R5 are within area 1. R6, R7, R8, R9,R10 are in area 2.

Figure 2 is the topology to test the Paramiko and Telnetlib libraries in EGP routing. This topology is composed of 3 US, namely AS 100, AS 200, and AS 300. For AS 100 contains R11, AS 200 contains R1,R2,R3,R4 and R5. AS 300 it contains R6, R7,R8, R9 and R10. Eleven routers here are used to receive IGP and EGP routing configurations provided by Network Automation Ubuntu. By providing this routing configuration, performance results can be obtained from the Paramiko and Telnetlib libraries for network automation. Client PCs 1 and 2 function to transmit data. This data transmission is in the form of ICMP packets that determine whether client PCs 1 and 2 are connected through IGP and EGP configurations. In this topology, IP configuration is needed to give identity to each interface on the client PC, router, and Ubuntu Network Automation.

2.2 Testing and Data Retrieval

Tests were conducted to obtain data from the performance of the Paramiko and Telnetlib libraries. The data needed in this study is in the form of configuration delivery time to the router, IGP, and EGP network convergence time, as well as throughput and delay values. Testing is carried out 20 times per library.

2.2.1. Configuration Delivery Time to the Router

Configuration Delivery Time is measured by sending a configuration IP address and routing protocol IGP and EGP from the Ubuntu network automation system to each router. The author uses a laptop device to capture traffic on the path between Ubuntu network automation and switches via Wireshark software. The data captured in traffic capture is from SSH and Telnet protocols. Then, the data is obtained by calculating SSH and Telnet connections' beginning and end times on each router. The results of measuring the configuration delivery time to all routers in this parameter are seen from the library side.

2.2.2. Convergence time in IGP and EGP

Data retrieval convergence time is carried out because it determines whether the router can forward a packet to its destination. The router can route or forward packets if it already knows the route information stored in the routing table. If all route information is known, the network has entered a convergent state, and the router can forward the packet to the destination. The author used a laptop device to capture traffic on the path between Client 1 and Router 3 via Wireshark software. IGP convergence time data is obtained when Client 1 sends ICMP data packets or pings Client 2 with the initial "Destination Host Unreachable" condition. When the condition has been "replied," the condition has converged. The vulnerability between DHU and Reply conditions results from data convergence time.

2.2.3. Throughput dan Delay

Throughput is the value of a data rate measured in bps (bits per second). The following is the formula for calculating the throughput value[20].

$$Throughput : \frac{Dp}{Ld} \quad (1)$$

Information :

Dp = Total data send

Ld = Total packet received

The standardised classification of Throughput values based on TIPHON TR 101 329 V2.1.1 (1999-06) is described in table 1.

Table 1 Throughput categories

Index	Delay (ms)	Category
4	100	Very Good
3	75	Good
2	50	Moderate
1	<25	Poor

Delay is the time for data packets to reach their destination, delays are usually caused by queues or other routes[20].

$$Delay = \frac{\sum D}{Pr} \quad (2)$$

Information:

$\sum D$ = Total delay / Length of time to send data

Pr = Total packet received

The standardization classification of Delay values based on TIPHON TR 101 329 V2.1.1 (1999-06) is described in table 2.

Table 2 Delay categories

Index	Delay (ms)	Categories
4	<150	Very Good
3	150-300	Good
2	300-450	Moderate
1	>450	Poor

The last data retrieval process is QoS data retrieval in the form of Throughput and Delay. The author used a laptop device to capture traffic on the path between the switch and router via Wireshark software. The protocols captured in this parameter are the SSH and Telnet protocols. The data taken for QoS Delay is the total delay and packets received, then the data is calculated using the formula in equation (2). As for QoS throughput, which is the packet sent and the delivery time, the data is calculated

using the formula in equation (1). The QoS Throughput and Delay values will be obtained with the data calculated by the existing equation.

3. RESULT AND DISCUSSION

3.1. Configuration Delivery Time to the Router

Figure 4.1 measures the time for configuring IP addresses and IGP routing in the form of OSPF for each router for 20 attempts. The data is obtained by retrieving data traffic from Ubuntu network automation to switch devices. Protocol. The telnetlib library has an average value for giving IGP configuration time to the router of 59.194 seconds, while the Paramiko library has an average value of 77.119 seconds. So, the results obtained in the telnet lib library are faster than those obtained in the Paramiko library, with a time difference of 21.297 seconds.

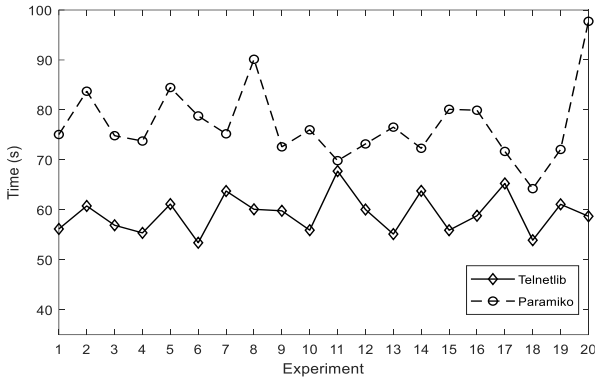

Figure 4. Results of IGP configuration Delivery Time to the router

Due to differences in remote protocols, Paramiko uses the SSH protocol, which has an encryption process, so that the data size is increasing, and the telnet lib library uses the telnet protocol, which has no encryption process. With the difference in protocols, the amount of data in each library differs, affecting the time to send IGP configurations to the router.

Table 3. Differences in the Number of Telnetlib and Paramiko Data Libraries on the IGP Routing Protocol

Library	Sum of data (byte)
Telnetlib	299591
Paramiko	534734

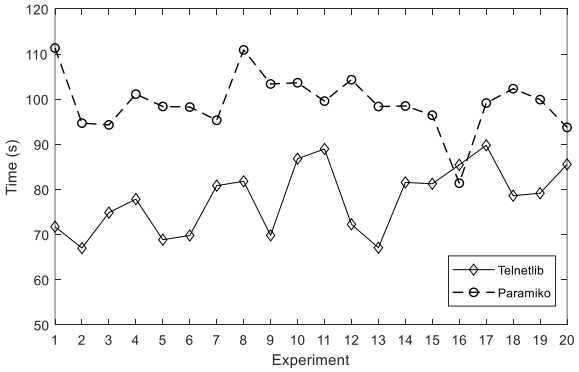

Figure 5. Results of IGP configuration Delivery Time to the router

Figure 5 measures the time spent giving IP address configuration and EGP routing in the form of BGP to each router 20 times. The Telnetlib library has an average value of 77.973 seconds for giving EGP configuration to the router, while the Paramiko library has a result of 99.270 seconds. As a result, the results obtained using the Telnetlib library are 21.297 seconds faster than those obtained using the Paramiko library. Similar to sending configuration in IGP routing, in EGP routing, the use of different protocols of the two libraries affects the delivery time. But on the routing protocol side, sending the IGP routing protocol is faster than EGP because the EGP routing process is more so that it makes the line in the Paramiko and Telnetlib script libraries larger.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Table 4. Differences in the Number of Telnetlib and Paramiko Data Libraries in EGP Routing Protocol

Library	Sum of data (byte)
Telnetlib	463447
Paramiko	828384

Research [5], [17] states that the paramiko library is faster than the netmiko library. When the Paramiko library is compared to telnetlib, the telnetlib library is faster than the Paramiko library. Although both libraries have the same characteristics, they have different remote protocols. This protocol difference affects the length of the sending time because telnetlib uses the telnet protocol, which does not have an encryption process for sending data. So that the data in the telnetlib library is smaller, as can be seen in tables 4 and 5, thus making the delivery time faster.

3.2. Convergence Time in IGP and EGP

IGP convergence time data is obtained when Client 1 sends ICMP data packets or "pings" Client 2 with initial conditions in the form of "Destination Host Unreachable" until the "reply" condition. Data retrieval is performed on the path between client one and R3.

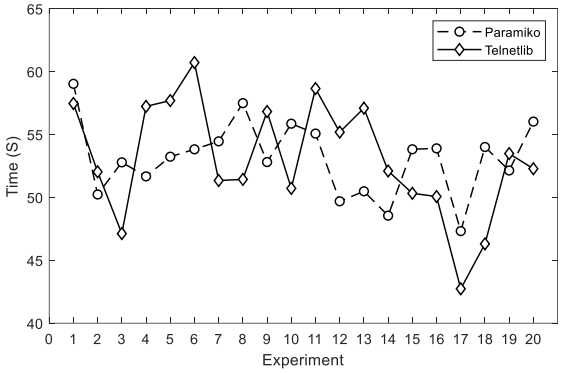

Figure 6. IGP Network Convergence Time

Figure 6 shows the IGP network convergence time using the telnetlib and Paramiko libraries. The Telnetlib library has an average IGP convergence time value of 53.042 seconds. As for the Paramiko library, it has an average convergence time value of 53.121 seconds. So, the results showed that the telnetlib library was faster than the Paramiko library, with a time difference of 0.079 seconds. The time the configuration is sent to the router also affects the convergence time. So, the convergence time is directly proportional to the time spent sending the configuration to the router.

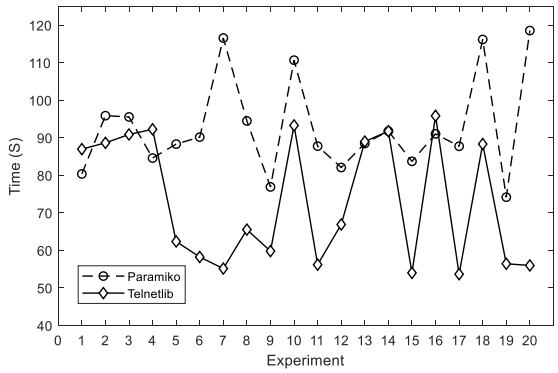

Figure 7. EGP Network Convergence Time

Figure 7 is the result of the convergence time of the EGP network using the telnetlib and Paramiko libraries. The telnetlib library has an average IGP convergence time value of 73.046 seconds. As for the Paramiko library, it has an average convergence time value of 92.736 seconds. The results of the EGP network convergence time can be seen in Figure 6. The results obtained by the telnetlib library are faster than those obtained by the Paramiko library, with a difference of 19.69 seconds. In terms of routing protocol, the IGP routing protocol is faster than the EGP. Because the IGP routing protocol is in the form

of OSPF, sending packet hello with a default time of 10 seconds. While the EGP or BGP routing protocol sends a Minimum Route Advertisement (MRAI) with a default time of 30 seconds.

3.3. Throughput

Measurement The throughput value is based on capturing SSH and Telnet traffic using Wireshark on switch and router lines. The measurement is done while the configuration is transmitted by Ubuntu network automation to the router.

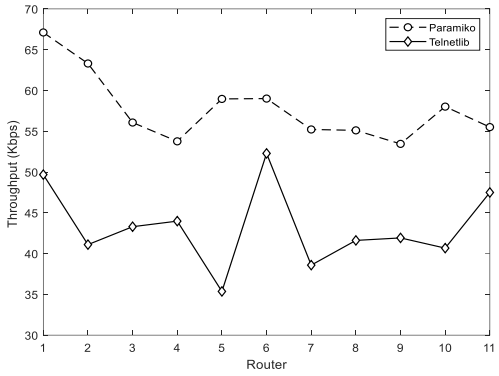

Figure 8. IGP Library Paramiko and Telnetlib Throughput Results

Figure 8 is the IGP throughput result from the Paramiko and Telnetlib libraries. Paramiko libraries have an average throughput value of 57,774 kbps. Meanwhile, the telnetlib library has an average throughput value of 43,270 kbps. So, it was determined that the Paramiko library has a better throughput value than the telnetlib library, with a difference of 14,504 kbps. The throughput value of Paramiko is greater because the amount of data sent to the router is greater than using the Telnetlib library.

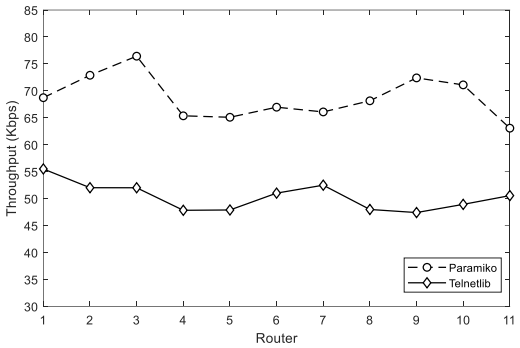

Figure 9. IGP Library Paramiko and Telnetlib Throughput Results

Figure 9 shows EGP throughput from the Paramiko and Telnetlib libraries. Paramiko libraries have an average throughput value of 68,757 kbps. Meanwhile, the telnetlib library has an average throughput value of 50,335 kbps. These results determined that the Paramiko library has a better throughput value than the Telnetlib library, with a difference of 18,442 kbps. The size of this data is influenced by the number of program lines in each library, and for Paramiko data libraries, more lines are added with the encryption process.

In terms of routing protocol, the EGP routing protocol has a greater throughput value. When sending EGP configurations using both libraries, the total amount of data is greater than IGP can be seen in Table 9.

3.4. Delay

Measurement The throughput value is based on capturing SSH and Telnet traffic using Wireshark on switch and router lines. The measurement is done while the configuration is transmitted by Ubuntu network automation to the router.

p-ISSN: 2528-1682
e-ISSN: 2527-9165

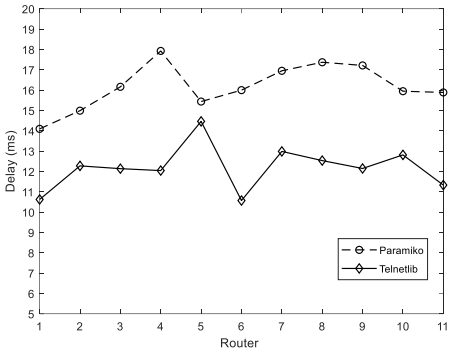

Figure 10. IGP Library Paramiko and Telnetlib Delay Results

Figure 10 shows IGP delay measurement results using the Paramiko and Telnetlib libraries. The telnetlib library has an average delay value of 12.174 seconds, while the Paramiko library has an average delay value of 16.179 seconds. So, the results of the Telnetlib library have a smaller or better delay than the Paramiko library, with a difference in value of 4.005 ms.

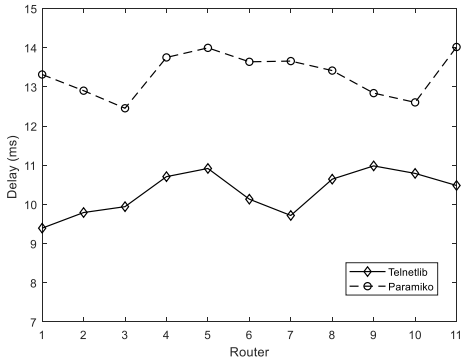

Figure 11. EGP Library Paramiko and Telnetlib Delay Results

Figure 11 results in measuring EGP delay using the Paramiko and telnetlib libraries. The Telnetlib library has an average delay value of 10.484 seconds, while the Paramiko library has an average delay value of 13.328 seconds. Thus, the results of the telnetlib library have a smaller or better delay than the Paramiko library, with a difference in value of 2.844 ms. The telnetlib library has a better delay than Paramiko because the sending time of IGP and EGP configurations to the telnetlib library router is faster than Paramiko. The size of the delay value is affected by the time parameter for sending the configuration to the router.

In terms of routing protocol, the EGP routing protocol has a greater throughput value. When sending EGP configurations using both libraries, the total amount of data is greater than IGP can be seen in Table 5.

Table 5 Amount of data, packet, and sending time of IGP and EGP

Routing Protocols	**Paramiko**		**Telnetlib**	
	Data(*bytes*)	**Time(s)**	**Data(*bytes*)**	**Time(s)**
IGP	532826	37,375	298055	65,424
EGP	830308	42,461	463717	70,025

4. CONCLUSION

Based on the results of network automation testing using Paramiko and Telnetlib libraries on IGP and EGP routing protocols, it can be concluded that Telnetlib libraries are better than Paramiko libraries in configuration delivery time parameters of 23,361%, convergence time of 10.69%, and delay of 23,661% when applied to IGP and EGP routing protocols. The Paramiko library is better than Telnetlib in the QoS parameter in the form of 35.005% throughput when applied to IGP and EGP routing protocols.

Unlike the previous study [5], [17], the telnetlib library is faster than the Paramiko library in this study due to differences in the protocols used by the two libraries. However, the telnetlib library

delivers time, configuration, and convergence faster. However, the library needs to improve, namely, the lack of security in sending data because there is no encryption process on the Telnet protocol.

REFERENCES

[1] S. Nugroho, B. Pujiarto, U. M. Magelang, and P. Korespondensi, "Network Automation Pada Beberapa Perangkat Router Network Automation in Some Router Devices," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 1, pp. 79–86, 2022, doi: 10.25126/jtiik.202293947.

[2] H. Saptono, A. Rustianto, T. Informatika, and S. Tinggi Teknologi Terpadu Nurul Fikri Jakarta Selatan, "Jurnal Informatika Terpadu ANALISIS TINGKAT EFISIENSI PADA KONFIGURASI MIKROTIK HOTSPOT MENGGUNAKAN METODE ZERO TOUCH PROVISIONING," *J. Inform. Terpadu*, vol. 7, no. 2, pp. 47–52, 2021, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[3] G. S. Santyadiputra, I. M. E. Listartha, and G. A. J. Saskara, "The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Mar. 2021. doi: 10.1088/1742-6596/1810/1/012028.

[4] L. Wijaya and A. B. Silviana, "Aplikasi Otomatisasi Jaringan Berbasis Command Line Interface Pada Router Cisco Dan Mikrotik," *ICIT J.*, vol. 8, no. 2, pp. 158–171, 2022, doi: 10.33050/icit.v8i2.2406.

[5] L. G. Mauboy and T. Wellem, "Studi Perbandingan Library Untuk Implementasi Network Automation Menggunakan Paramiko Dan Netmiko Pada Router Mikrotik," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 4, pp. 790–799, 2022, doi: 10.30865/jurikom.v9i4.4420.

[6] E. S. Ginting, S. Suroso, and I. Hadi, "Pengujian Konfigurasi Otomatis Penambahan Gateway Pada Virtual Router Menggunakan Aplikasi Otomatisasi Jaringan Berbasis Web," *J. Media Inform. Budidarma*, vol. 4, no. 4, pp. 1126–1131, 2020, doi: 10.30865/mib.v4i4.2485.

[7] George Milios, "Network Automation Using Python," *NetworkCOm*, no. December, pp. 1–15, 2020, [Online]. Available: https://repository.ihu.edu.gr/xmlui/bitstream/handle/11544/29802/Network Automation using Python%28final%29.pdf?sequence=1

[8] A. M. Mazin, R. A. Rahman, M. Kassim, and A. R. Mahmud, "Performance analysis on network automation interaction with network devices using python," in *ISCAIE 2021 - IEEE 11th Symposium on Computer Applications and Industrial Electronics*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 360–366. doi: 10.1109/ISCAIE51753.2021.9431823.

[9] "Network Automation (Otomasi Jaringan) | by Skudou | Medium." https://bagiinterest.medium.com/network-automation-otomasi-jaringan-115ef45b45ac (accessed Nov. 22, 2022).

[10] Hamzan Wadi.ST, *Pemrograman Python untuk Pelajar dan Mahasiswa*. TR Publisher.

[11] S. Maruch and A. Maruch, *Python For Dummies*, vol. 2006. 2006.

[12] K. Byers, "Python for Network Engineers | Articles," *Https://Pynet.Twb-Tech.Com/Blog*, 2015.

[13] T. Peters, "MASTERING PYTHON NETWORK AUTOMATION."

[14] Ahmad Rosid Komarudin, *Otomatisasi Administrasi Jaringan Dengan Script Python*, Cetakan Pertama. Jasakom., 2018, 2018.

[15] Huawei Technologies Co., Ltd., *Data Communications and Network Technologies*. Springer Nature Singapore, 2023. doi: 10.1007/978-981-19-3029-4.

[16] P. Kathiravelu and M. O. F. Sarker, *Python network programming cookbook : overcome real-world networking challenges*.

[17] K. Nugroho, A. D. Abrariansyah, and S. Ikhwan, "Perbandingan Kinerja Library Paramiko dan Netmiko dalam Proses Otomasi Jaringan," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 1–8, 2020.

[18] M.-I. Candrea-Bogza and P. Ciotîrnae, "Integrated Management of Transport and Commutation Resources over the Network Layer," *J. Mil. Technol.*, vol. 2, no. 1, pp. 27–30, Jun. 2019, doi: 10.32754/JMT.2019.1.05.

[19] R. A. Wiryawan and N. R. Rosyid, "Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python," *Simetris*, vol. 10, no. 2, pp. 1–12, 2019.

[20] ETSI, "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)," *Etsi Tr 101 329 V2.1.1*, vol. 1, pp. 1–37, 2020.