Indonesian Journal of Electrical Engineering and Computer Science
Vol. 24, No. 1, October 2021, pp. 329~337
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v24.i1.pp329-337

Analysis of IPv6 jumbogram packages transmission using jumbo frame in mikrotik-based tunneling

Yahya Hidayatullah[1], Arief Marwanto[2], Imam Much Ibnu Subroto[3]
[1,2]Department of Electrical Engineering, Universitas Islam Sultan Agung (UNISSULA), Semarang, Indonesia
[3]Department of Information Technology and Computer Science, Universitas Islam Sultan Agung (UNISSULA) Semarang, Indonesia

Article Info

Article history:

Received Sep 19, 2020
Revised Jun 25, 2021
Accepted Jul 15, 2021

Keywords:

Generator traffic
IPv6
Jumbo frame
Operation systems
Tunnelling

ABSTRACT

The validation and accuracy of internet protocol version 6 (IPv6) performance using jumbo frames is still not perfect, due to peer-to-peer connections testing within the same operating system and between operating systems. Therefore, inaccurate data test results. To mitigated, testing with a wider platform is recommended, a medium-scale network connection is proposed such as metropolitan area networks. In this works, a connection between computer devices connected by three proxy routers are made, with different IPv6 segments on each port. Then each computer device sends traffic data to each other using a traffic-generator application. The first test through three routers without tunnel connection is carried out as the first scenario to compare performance with tunnel-based testing. Three parameters have been used in this test, such as maximum transfer unit (MTU) 1500 bytes, MTU 400 bytes and MTU 9000 bytes. The results of the tests conducted show that the use of jumbo frames using a proxy is less effective, even though it produces a larger throughput when using the MTU 4000, but there is fragmentation in the packet passing through the proxy because the packet passing through the proxy is split into 1500 byte sizes.

Corresponding Author:

Arief Marwanto
Department of Electrical Engineering
Universitas Islam Sultan Agung (UNISSULA)
Raya Kaligawe St. Km 04, Semarang 50112, Indonesia
Email: arief@unissula.ac.id

1. INTRODUCTION

Jumbo frame data transmission utilization in internet protocol version 6 (IPv6) research is less attention [1]-[3]. Moreover, most researchers are focuses on round trip time (RTT) a number of 1500 bytes of maximum transmission unit (MTU) during sending process in Windows and Linux platform [4]-[11]. Previous research [12]-[19] was conducted only comparing round trip time (RTT) and data transmission time using file transfer protocol (FTP). A normal maximum transfer unit (MTU) of 1500 bytes is used and a modified MTU of up to 9000 bytes is also used by some researchers. Jumbo frame is a frame size sent from ethernet with a payload size exceeding 1500 bytes, while gigabyte Ethernet is a jumbo frame transmission ethernet that can transmit 1 gigabyte of data in one second [20]-[27].

In [28] The purpose of this research is to find out how the performance is obtained when using jumbo frames. The method is performed by retrieving files from the ftp server. The results obtained by the network throughput using a larger jumbo frame so that the data retrieval time is twice as fast. In other hand, [29] has proposed examination of jumbo frames in similar operating systems. The purpose of this study

was to see the performance when using jumbo frames. The method used is sending data between 2 computers with the same operating system. By using jumbo frames, data traffic is expected to be more stable [30]-[36]. Peer-to-peer performace testing has been proposed by [37] with Windows and Linux platform without passing some sort of connections from couple router hardwares.

A few jumbo frame tests are conducted by [38]-[47] using the tunnel method only peer-to-peer and not testing on a true scale. Testing in the use of home and office computers, or computers with general specifications that are widely used in the community has not been widely used. This is because no one has mentioned that jumbo frames can run on tunnel media. Meanwhile, for communication between two or more offices that require a large data connection [48]-[52], tunnel media can be used by maximizing the running troughput. by increasing the MTU value on the tunnel connection, the number of packets sent is reduced so that the network structure that will be carried out in the test is to build a new network, consisting of 4 computers connected by 3 router units with different network addresses, then the router will be arranged so that the connection between computer running using tunnel media. There are three parameters used in testing this jumbo frame, namely MTU 1500 bytes, MTU 400 bytes and MTU 9000 bytes, and MTU parameters above 9000 bytes cannot be performed, because the device is not capable, then forward the connection made is using the transmission control protocol (TCP) protocol.

2. RESEARCH PROPOSED METHOD

In this work, network modeling will be carried out by building a seven-node connection using three routers and four computers, at the end of the network there will be a computer with Windows and Linux operating systems. At the time of testing, data traffic will be captured using the Wireshark application, with this application it will be seen how many troughputs are running, how many packets are sent, how many packets are lost and also the jitter that occurs during transmission.

The testing technique is that there are two computer devices connected to a computer network, connected by three routers will send data using a traffic-generator application called iperf, and run for 10 seconds on each test, the protocol used in the throughput test is TCP computers that are used using Windows and Linux operating systems. The testing is done by sending data from a computer with the same operating system, then across operating systems. The network structure used uses 2 methods, the first is the network that is connected directly using a cable on each device, as shown in Figure 1.

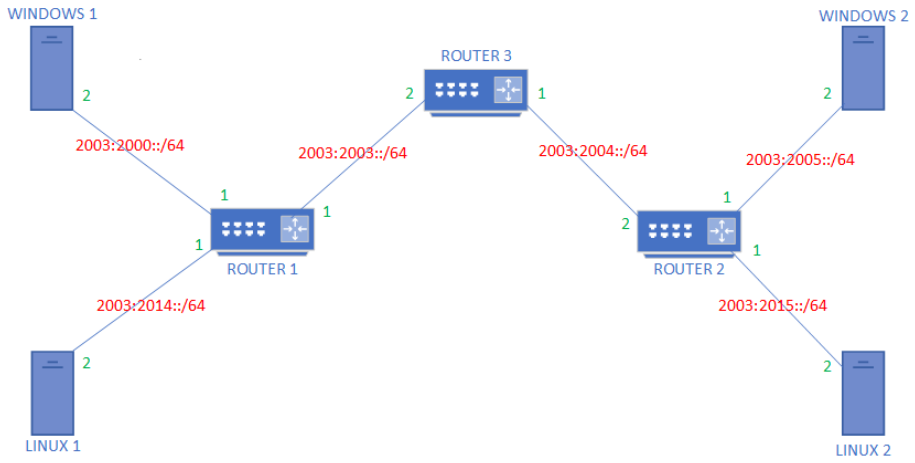

Figure 1. Direct connection without tunnel media

Then the second network method, the connection is made using tunneling on the router device, so that routers 1 and 2 are directly connected using the tunnel method. As shown in Figure 2. Technical testing is done like when testing a connection without using a tunnel, namely by sending traffic between 2 computers, for the type of tunnel used is IP IPv6 tunnel, by encapsulating the IP packet in the IP packet then sending it into the network to another router by using IPv6 [53]-[57].

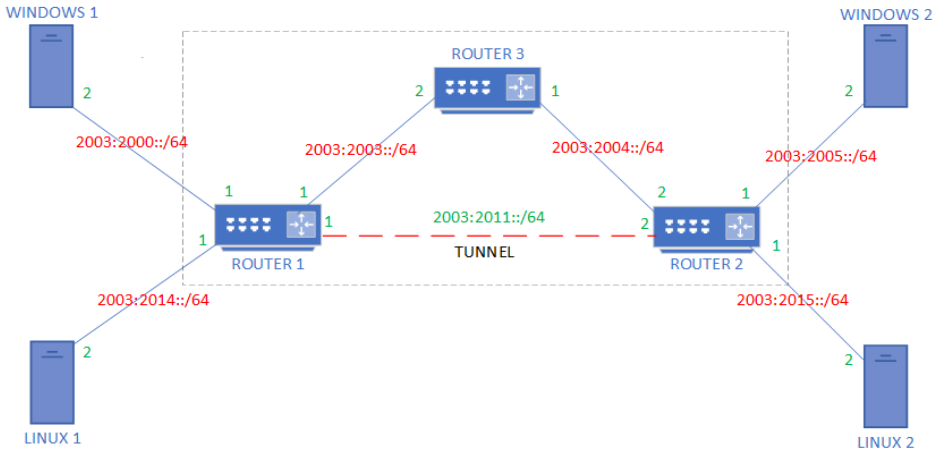

Figure 2. Connection using a tunnel

3. RESULTS AND ANALYSIS

Connection testing between the same operating systems and between operating systems is carried out to measure network performance using jumbo frames. There are two tests, namely testing without using a tunnel and testing using a tunnel.

3.1. The performance without tunnel

Tests with the same Windows operating system have been carried out using the Iperf application for 10 seconds, then Wireshark captures the data sent and received after Iperf runs for 10 seconds, as shown in Figure 3, the data that has been recorded by Wireshark. In Figure 3, the results of testing using a computer with the Windows operating system are shown. With an MTU of 4000 bytes, the resulting throughput is the MTU with the largest throughput of 914 Mbps. Tests show that the greater the MTU used, the fewer packets sent for the number of packets sent for 10 seconds. When using MTU 1500 bytes, the total number of packets sent is 633,785 packets, and if MTU 4000 bytes is used, it can send 288,580 packets and MTU 9000 bytes has sent 121,745 packets.

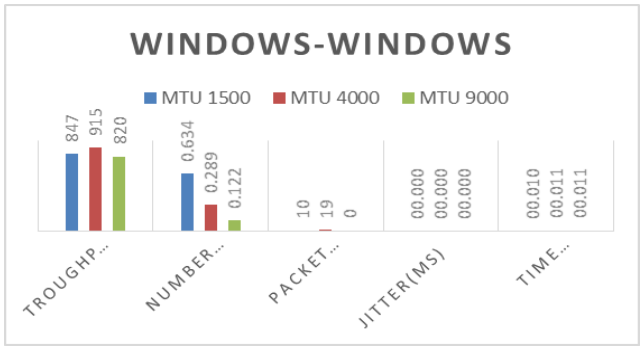

Figure 3. Testing Windows without tunnels

In Windows testing, when using MTU 4000 bytes, the most packet loss occurred, there're only 19 packet loss. Furthermore, the jitter test that occurs during testing shows that the larger the packet sent, the greater the jitter values. Likewise, the time span has also shown results that are not different from jitter, the larger the packet size sent, the longer it takes.

In Figure 4, the test on the Linux operating system shows that the fellow Linux operating system platforms are not able to send packets with an MTU of more than 9000 bytes. So that the use of MTU 4000 bytes has shown results with the greatest throughput, which is equal to 955 Mbps. The number of packets that can be sent is also getting less, for MTU 9000 bytes the packet sent is the value of the packet that was sent from ethernet but failed to send so the value is very small. In testing Linux and Linux, MTU 4000 bytes shows that the test does not produce a packet loss greater than MTU 1500 bytes, less packet loss that occurs

only 4 packets. however, the jitter at the 4000 bytes MTU still occurs and shows a higher number than the 1500 bytes MTU, with a value of 0.0374 ms.

The use of the 4000 bytes MTU time span has shown that in addition to producing a larger throughput, the transmission time can also be faster, so that it is unable to send packets with an MTU of more than 9000 bytes. So that the use of MTU 4000 has shown results with the greatest throughput, which is equal to 955 Mbps. In testing Linux to Linux, the use of 4000 bytes MTU did not result in a packet loss greater than MTU 1500 bytes, the packet loss was less, only 4 packets. In the use of MTU 4000 bytes, jitter still occurs and still shows a higher number than MTU 1500 bytes, which is equal to the value of 0.0374 ms. Meanwhile, the time span shows that using an MTU of 4000 bytes, in addition to producing a larger throughput, data transmission has also shown that the transmission time is also faster.

Figure 5 shows the results of cross-operating systems testing with the test method without using a tunnel between the Linux and Windows operating systems. The results of these tests show that the largest throughput generated is MTU 4000 bytes, the number of packets sent for MTU use is 9000 bytes, namely 126,167 packages per test which is the least packet delivery. The greater the MTU used, the less packet loss that occurs, by using an MTU of 9000 bytes, packet loss does not occur. However, the jitter that occurs is that the greater the MTU the greater the jitter that occurs. And the time span also shows results like jitter, where the greater the MTU, the greater the span that occurs.

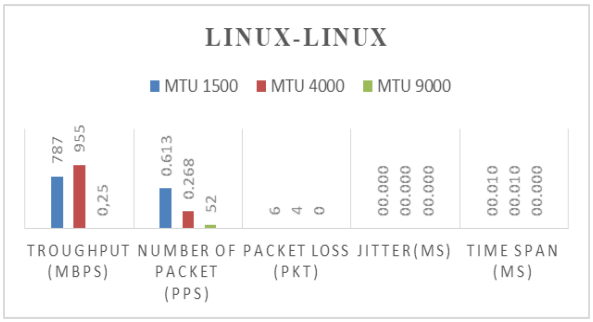

Figure 4. Testing Linux-Linux without tunnels

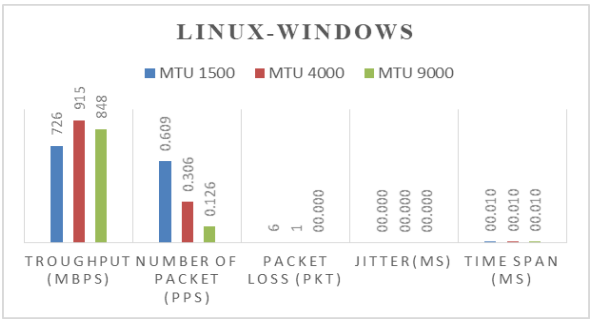

Figure 5. Linux-Windows testing without tunnels

3.2. Tunneling testing

The second stage of testing is carried out using a network model that has a tunnel link in the connection between the routers used. Three tests were carried out, the first and second tests used the same operating system, then the third test was cross-testing between Windows and Linux operating systems. Figure 6 has shown the performance testing of Windows-Windows via tunnel. The results of the analysis are.

In Figure 6, the tunnel network is used in the first test on the same Windows operating system. The test results show that for the greatest throughput is to use the MTU 4000 bytes. Meanwhile, the smallest total number of packets when sent in one test using MTU is 9000 bytes, which is 106,494 packages. There were only 3 packet loss that occurred during the test, when using MTU 4000 bytes. Then for jitter shows that the larger the packet sent, the greater the resulting jitter. When using the 4000 bytes MTU, the time span shows the fastest time among other MTUs, it is at 10.2026 seconds.

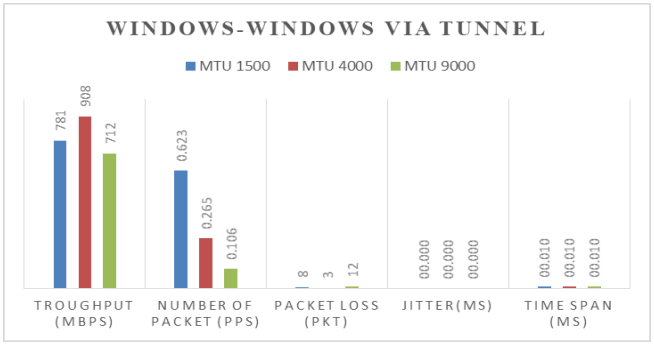

Figure 6. Tunneling Windows to Windows testing

In the next test is testing on the same Linux operating system, as shown in Figure 7. The test results show that the largest throughput is the one using the MTU 4000 bytes, while the 9000 bytes MTU shows that the system cannot send data, because the operating system is unable sends a packet larger than the MTU of 900 bytes. Figure 7 shows that the number of packets sent using the MTU of 4000 bytes is a total of 299,877 packets. The difference of 1 packet loss is shown in the use of MTU 1500 bytes and MTU 9000 bytes. The jitter that occurs is still greater for the use of a larger MTU as well. Meanwhile, the time span at MTU 4000 bytes has shown a value smaller than MTU 1500 bytes.

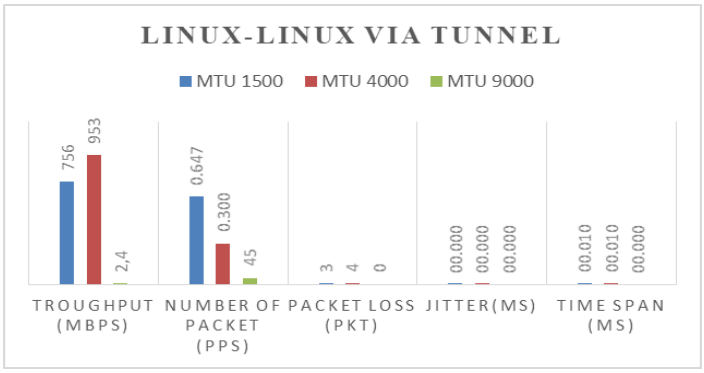

Figure 7. Linux tunneling to Linux testing

The last test was carried out between cross-platform operating systems, Windows and Linux as shown in Figure 8. The test results show that the throughput can run optimally using the MTU 4000 bytes, and the total number of packets generated in one test using the 4000 bytes MTU is 264,943 packages. While packet loss using MTU 4000 bytes in this test does not occur, for the jitter is still the same as the previous test results, with the use of a larger MTU it will produce a greater jitter, and the resulting time span shows that in the use of 9000 bytes MTU , this test has shown the lowest value.

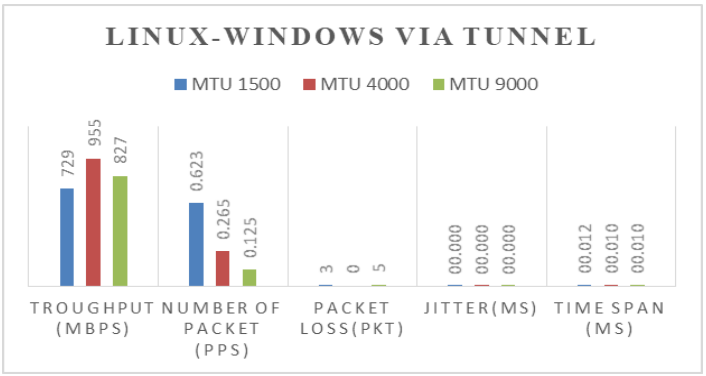

Figure 8. Tunneling Linux to Windows testing

3.3. Traffic on mikrotik

Retrieval of traffic data that passes through the mikrotik is not in accordance with the MTU that is set, on a mikrotik packet size that is more than 1500 bytes will be fragmented into small packets with a size of 1500 bytes illustrated in Figure 9. In Figure 9 shows the results of the tests that have been carried out, computer 1 sends a packet to computer 2 going through several mikrotik routers, the jumbo size of the packet that comes out of computer 1 is received by router 1, then from router 1 the packet is fragmented into a small size, then sent to to router 3 and proceed to router 2, after the packet is received by router 2, the size of the packet which was previously broken down into small pieces is returned back to its original size and continues sending the packet to computer 2, the resend packet and lost segmen will show in Figure 10. In Figure 10 shows the sending of jumbo packets between mikrotik, packet fragmentation occurs, the fragmentation process causes packet loss between mikrotiks so that performance and throughput decrease.

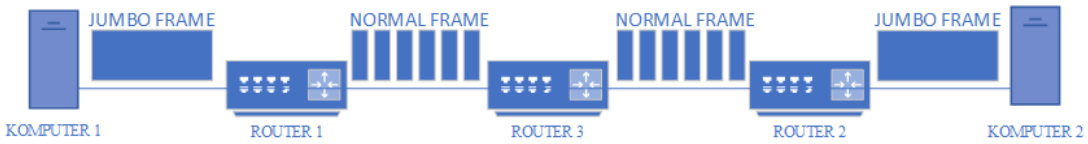

Figure 9. Package fragmentation on the mikrotik

1251 3.145320	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51118118 Ack=1 Win=212992 Len=8940
1252 3.145447	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51135998 Ack=1 Win=212992 Len=8940
1253 3.145518	2003:2000::2	TCP	9014 49747 → 5201 [ACK] Seq=51144938 Ack=1 Win=212992 Len=8940
1254 3.145653	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51162818 Ack=1 Win=212992 Len=8940
1255 3.145713	2003:2005::2	TCP	74 [TCP ACKed unseen segment] 5201 → 49747 [ACK] Seq=1 Ack=51103266 Win=212992 Len=0
1256 3.145885	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51189638 Ack=1 Win=212992 Len=8940
1257 3.145979	2003:2005::2	TCP	74 5201 → 49747 [ACK] Seq=1 Ack=51127058 Win=212992 Len=0
1258 3.146169	2003:2005::2	TCP	74 5201 → 49747 [ACK] Seq=1 Ack=51153878 Win=212992 Len=0
1259 3.146337	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51234338 Ack=1 Win=212992 Len=8940
1260 3.146515	2003:2005::2	TCP	74 [TCP ACKed unseen segment] 5201 → 49747 [ACK] Seq=1 Ack=51198578 Win=212992 Len=0
1261 3.146715	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51249190 Ack=1 Win=212992 Len=8940
1262 3.146732	2003:2000::2	TCP	9014 [TCP Retransmission] 49747 → 5201 [ACK] Seq=51249190 Ack=1 Win=212992 Len=8940
1263 3.146862	2003:2000::2	TCP	9014 [TCP Previous segment not captured] 49747 → 5201 [ACK] Seq=51267070 Ack=1 Win=212992 Len=8940

Figure 10. Sending packet between mikrotik

4. CONCLUSSION

Tests carried out between computer devices using a network system without using a tunnel and using a tunnel has been done. The test results show that the use of 4000 bytes MTU can produce the greatest throughput in sending data on the same operating system and across operating systems. The number of packages sent indicates that the greater the MTU used, the fewer the number of packages sent. So this 4000 bytes MTU is ideal for running on the IPv6 platform. The amount of packet loss that occurs during the test does not always indicate that the use of a large MTU is bound to occur with a large packet loss, or vice versa. The occurrence of packet loss depends on the readiness of all connected devices, if there are devices that are not ready to receive the packet, then packet loss will occur during transmission.

The data shows that the greater the MTU used, the greater the jitter value and Jitter occurs because it is influenced by the amount of MTU used. The use of a larger MTU shows that it can reduce the time span value, so that performance is better. The analogy in using jumbo frames in everyday life is like sending goods by entering into a larger size box, when the box size is small, the shipping process will take longer, whereas by using a larger box, the number of goods sent can be reduced, so that the shipping process can run faster, but increasing the size of the box does not always make the delivery of goods faster, enlarging the shipping medium must be adjusted to the maximum capacity of the transport medium, because sending packages with sizes that exceed the capacity or are too large can also reduce performance when delivery of goods.

The use of jumbo frames using a mikrotik is less effective, even though it produces a larger throughput when using the MTU 4000, but there is fragmentation in the packets passing through the mikrotik because the packet passing through the mikrotik is split into 1500 byte sizes. In the future, from the results of the research conducted, it can still be developed to find better test values and accuracy, by using a tunnel method other than IPIPv6, then in network design it can combine from different test locations on a wider scale.

REFERENCES

[1] A. Salih, "An adaptive approach to detecting behavioural covert channels in IPv6," Nottingham Trent University, 2017.
[2] M. Leinonen, "Layer 2 ethernet communication tunneling possibilities in automation systems," 2017.
[3] G. Naci and D. Haji, "Dynamic Behavior Analysis of ANFIS Based MPPT Controller for Standalone Photovoltaic Systems," *Int. J. Renew. Energy Res.*, vol. 10, no. 1, pp. 101-108, 2020.
[4] X. Chen, H. Kim, J. M. Aman, W. Chang, M. Lee, and J. Rexford, "Measuring tcp round-trip time in the data plane," in *Proceedings of the Workshop on Secure Programmable Network Infrastructure*, 2020, pp. 35-41, doi: 10.1145/3405669.3405823.
[5] T. Kato, X. Yan, R. Yamamoto, and S. Ohzahata, "A Study on Round-trip Time Estimation from Unidirectional Packet Traces Using Different TCP Congestion Control Algorithms," *Int. J. Adv. Networks Serv.*, vol. 12, no 1-2, 2019.
[6] M. Zhang *et al.*, "Will TCP work in mmWave 5G cellular networks?," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 65-71, 2019, doi: 10.1109/MCOM.2018.1701370.

[7] D. H. Hagos, P. E. Engelstad, A. Yazid, and C. Griwodz, "A Deep Learning Approach to Dynamic Passive RTT Prediction Model for TCP," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, 2019, pp. 1-10, doi: 10.1109/IPCCC47392.2019.8958727.
[8] I. Maslouhi, E. M. Ar Reyouchi, K. Ghoumid, and B. Kaoutar, "Analysis of end-to-end packet delay for Internet of Things in wireless communications," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 9, pp. 338-343, 2018.
[9] J. Crichigno, Z. Csibi, E. Bou-Harb, and N. Ghani, "Impact of segment size and parallel streams on TCP BBR," in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, 2018, pp. 1-5, doi: 10.1109/TSP.2018.8441250.
[10] H. Ahmed, M. J. Arshad, S. Muhammad, S. Ahmad, and A. H. Zahid, "Queue length-based load balancing in data center networks," *Int. J. Commun. Syst.*, vol. 33, no. 14, p. e4472, 2020, doi: 10.1002/dac.4472.
[11] B. Preveze, "A novel method for performance improvement of slow start congestion control method in packet switched networks," *Uludağ Univ. J. Fac. Eng.*, vol. 24, no. 3, pp. 451-468, 2019, doi: 10.17482/uumfd.644387.
[12] S. Narayan and P. R. Lutui, "TCP/IP Jumbo Frames Network Performance Evaluation on A Test-bed Infrastructure," *Int. J. Wirel. Microw. Technol.*, vol. 2, no. 6, pp. 29-36, 2012, doi: 10.5815/ijwmt.2012.06.05.
[13] J. Hurwitz and W. Feng, "Initial end-to-end performance evaluation of 10-Gigabit Ethernet," in *11th Symposium on High Performance Interconnects, 2003. Proceedings.*, 2003, pp. 116-121, doi: 10.1109/CONECT.2003.1231487.
[14] P. Joubert, R. B. King, R. Neves, M. Russinovich, and J. M. Tracey, "High-Performance Memory-Based Web Servers: Kernel and User-Space Performance," in *USENIX Annual Technical Conference, General Track*, 2001, pp. 175-187.
[15] S. Praptodiyono, A. S. Pramudyo, and R. Sofhan, "The Impacts of Transmitting IPv6 Packets Using Jumbo Frame on The Network Performance," in *PROCEEDINGS The 4th International Conference on Basic Sciences 2018: The Development of Sciences and Technology in Improving Natural and Cultural Resources*, 2019.
[16] P. Gray and A. Betz, "Performance evaluation of copper-based gigabit ethernet interfaces," in *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.*, 2002, pp. 679-690.
[17] R. Sofhan, R. Fahrizal, and A. Osman, "Performance evaluation of IPv6 jumbogram packets transmission using jumbo frames," in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017, pp. 1-5, doi: 10.1109/EECSI.2017.8239188.
[18] Y. Wu, S. Kumar, and S.-J. Park, "Measurement and performance issues of transport protocols over 10áGbps high-speed optical networks," *Comput. Networks*, vol. 54, no. 3, pp. 475-488, 2010, doi: 10.1016/j.comnet.2009.09.017.
[19] Y. Wu, S. Kumar, and S.-J. Park, "On transport protocol performance measurement over 10gbps high speed optical networks," in *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, 2009, pp. 1-6, doi: 10.1109/ICCCN.2009.5235238.
[20] L. Valcarenghi, K. Kondepu, and P. Castoldi, "Time-versus size-based CPRI in Ethernet encapsulation for next generation reconfigurable fronthaul," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 9, pp. D64-D73, 2017, doi: 10.1364/JOCN.9.000D64.
[21] M. J. Christensen and T. Richter, "Achieving reliable UDP transmission at 10 Gb/s using BSD socket for data acquisition systems," *J. Instrum.*, vol. 15, no. 9, p. T09005, 2020.
[22] A. Custura, G. Fairhurst, and I. Learmonth, "Exploring usable Path MTU in the Internet," in *2018 Network Traffic Measurement and Analysis Conference (TMA)*, 2018, pp. 1-8, doi: 10.23919/TMA.2018.8506538.
[23] T. Lackorzynski, G. Garten, J. S. Huster, S. Kopsell, and H. Hartig, "Enabling and Optimizing MACsec for Industrial Environments," *IEEE Trans. Ind. Informatics*, vol. 17, no. 11, 2020, doi: 10.1109/TII.2020.3040966.
[24] D. Wang and B. Hua, "ZCopy-Vhost: Replacing Data Copy With Page Remapping in Virtual Packet I/O," *IEEE Access*, vol. 7, pp. 51047-51057, 2019, doi: 10.1109/ACCESS.2019.2911905.
[25] N. M. Garcia, "The Internet Protocol--Past, some current limitations and a glimpse of a possible future," *arXiv Prepr. arXiv2104.01406*, 2021.
[26] T. Järvi, "Layer 2 Solutions in Access Provider Networks," 2020.
[27] Y. Song, Y. Wang, P. Bull, and J. D. Reiss, "Performance evaluation of a new flexible time division multiplexing protocol on mixed traffic types," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, 2017, pp. 23-30, doi: 10.1109/AINA.2017.80.
[28] D. Salyers, Y. Jiang, A. Striegel, and C. Poellabauer, "JumboGen: dynamic jumbo frame generation for network performance scalability," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 5, 2007, pp. 53–64.
[29] Supriyanto, R. Sofhan, R. Fahrizal, and A. Osman, "Performance evaluation of IPv6 Jumbogram Packets transmission using Jumbo Frames," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2017-Decem, no. September, pp. 19-21, 2017, doi: 10.1109/EECSI.2017.8239188.
[30] G. Hernandez-Oregon *et al.*, "Performance analysis of V2V and V2I LiFi communication systems in traffic lights," *Wirel. Commun. Mob. Comput.*, vol. 2019, 2019, doi: 10.1155/2019/4279683.
[31] Y. Liu, A. Dowling, and L. Huie, "Benchmarking Network Performance in Named Data Networking (NDN)," in *2020 29th Wireless and Optical Communications Conference (WOCC)*, 2020, pp. 1-6, doi: 10.1109/WOCC48579.2020.9114910.
[32] T. Lucas, M. Ferreira, R. Plachta, G. Ferreira, and K. Costa, "Non-Fragmented Network Flow Design Analysis: Comparison IPv4 with IPv6 Using Path MTU Discovery," *Computers*, vol. 9, no. 2, p. 54, 2020.
[33] J. Garay, J. Unzilla, E. Jacob, and J. Matias, "A Layer 2 Traffic Steering Mechanism for Network Slicing scenarios," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 384-392, doi: 10.1109/NETSOFT.2018.8460100.
[34] M. Wyss, G. Giuliari, M. Legner, and A. Perrig, "Secure and Scalable QoS for Critical Applications," *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, doi:

10.1109/IWQOS52092.2021.9521315.
[35] T. Hirai and T. Murase, "Node clustering communication method with member data estimation to improve qos of v2x communications for driving assistance with crash warning," *IEEE Access*, vol. 7, pp. 37691-37707, 2019, doi: 10.1109/ACCESS.2019.2906392.
[36] I. A. I. Diyeb, A. Saif, and N. A. Al-Shaibany, "Ethical network surveillance using packet sniffing tools: A comparative study," *Int. J. Comput. Netw. Inf. Secur.*, vol. 11, no. 7, pp. 12-22, 2018, doi: 10.5815/ijcnis.2018.07.02.
[37] S. Praptodiyono, R. Sofhan, A. S. Pramudyo, T. Firmansyah, and A. Osman, "Performance comparison of transmitting jumbo frame on Windows and Linux system," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 1, pp. 68-75, 2019, doi: 10.12928/telkomnika.v17i1.11627.
[38] A. Uta *et al.*, "Is big data performance reproducible in modern cloud networks?," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, 2020, pp. 513–527.
[39] H. Raj and K. Schwan, "High performance and scalable I/O virtualization via self-virtualized devices," in *Proceedings of the 16th international symposium on High performance distributed computing*, 2007, pp. 179-188, doi: 10.1145/1272366.1272390.
[40] M. G. Iatrou, A. G. Voyiatzis, and D. N. Serpanos, "Network Stack Optimization for Improved IPsec Performance on Linux," in *SECRYPT*, 2009, pp. 83-91.
[41] A. Zakari, M. Musa, G. Bekaroo, S. A. Bala, I. A. T. Hashem, and S. Hakak, "IPv4 and IPv6 protocols: A Comparative performance study," in *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*, 2019, pp. 1-4, doi: 10.1109/ICSGRC.2019.8837050.
[42] T. Koutny and J. Sykora, "Lessons learned on enhancing performance of networking applications by IP tunneling through Active Networks," *Int. J. Adv. Internet Technol.*, vol. 3, no. 3-4, 2010.
[43] I. Kotuliak, P. Rybár, and P. Trúchly, "Performance comparison of IPsec and TLS based VPN technologies," in *2011 9th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2011, pp. 217-221, doi: 10.1109/ICETA.2011.6112567.
[44] B. Hoekstra, D. Musulin, and J. J. Keijser, "Comparing TCP performance of tunneled and non-tunneled traffic using OpenVPN," *Univ. Van Amsterdam, Syst. Netw. Eng. Amsterdam*, pp. 2010-2011, 2011.
[45] Z. A. Zardari, M. Ali, R. A. Shah, and L. H. Zardari, "A Hybrid Technique for Tunneling Mechanism of IPv6 using Teredo and 6RD to Enhance the Network Performance," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 11, pp. 100-105, 2018, doi: 10.14569/IJACSA.2018.091115.
[46] W. Feng, P. Balaji, C. Baron, L. N. Bhuyan, and D. K. Panda, "Performance characterization of a 10-Gigabit Ethernet TOE," in *13th Symposium on High Performance Interconnects (HOTI'05)*, 2005, pp. 58-63, doi: 10.1109/CONECT.2005.30.
[47] J. Sykora and T. Koutny, "Enhancing Performance of Networking Applications by IP Tunneling through Active Networks," in *2010 Ninth International Conference on Networks*, 2010, pp. 361-364, doi: 10.1109/ICN.2010.64.
[48] Y. A. Makeri, "Design and Implementation of optimized features in a local area network for improvedenterprisenetwork," *GSJ*, vol. 7, no. 7, 2019
[49] D. Yun, C. Q. Wu, N. S. V Rao, and R. Kettimuthu, "Advising big data transfer over dedicated connections based on profiling optimization*," IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2280–2293, 2019.
[50] S. Kucera, K. Fahmi, and H. Claussen, "Latency as a service: Enabling reliable data delivery over multiple unreliable wireless links," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.
[51] A. H. A. Abubakar, A. Muhammad, and K. Mohammed, "CST4500: Computer Network And Internetworking." December 2020.
[52] M. I. Nashiruddin and N. Solihah, "Performance Evaluation of XGS-PON Optical Network Termination for Enterprise Customer," in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2020, pp. 251–256.
[53] I. Haryono, D. W. Sudiharto, and A. G. Putrada, "QoS Improvement Analysis of VoIP Service Which Uses Overlay Network. Case Study: Calling AWS VoIP Gateway From Bandung, Indonesia," in *2018 International Seminar on Application for Technology of Information and Communication*, 2018, pp. 381–387.
[54] S. T. Aung and T. Thein, "Comparative Analysis of Site-to-Site Layer 2 Virtual Private Networks," in *2020 IEEE Conference on Computer Applications (ICCA)*, 2020, pp. 1–5.
[55] Y. K. Sharma and C. Kaur, "The Vital Role of Virtual Private Network (VPN) in Making Secure Connection Over Internet World*."International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, March 2020.
[56] S. Mackey, I. Mihov, A. Nosenko, F. Vega, and Y. Cheng, "A Performance Comparison of WireGuard and OpenVPN," in *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020, pp. 162–164.
[57] R. Angelo, "Secure Protocols and Virtual Private Networks: An Evaluation.," *Issues Inf. Syst.*, vol. 20, no. 3, 2019.

BIOGRAPHIES OF AUTHORS

Yahya Hidayatullah was born in Semarang, 02 Sseptember 1990, obtained an Associate Degree (A.md) in Telecommunication Engineering from Semarang State Polytechnic, and graduated in 2015. Obtained a Bachelor of Engineering (ST), Department of Informatics, Sultan Agung Islamic University (UNISSULA) Semarang, graduated in 2016. He's finished study in Master of Electerical Engineering from Universitas Islam Sultan Agung (UNISSULA) Semarang in 2020. Currently a Director of CV. Maudy Communication Network in Semarang.

Arief marwanto, ST, M. Eng, Ph. D is lecturer in Master of Electrical Engineering Department, Faculty of Industrial Technology, Universitas Islam Sultan Agung (UNISSULA), Semarang, Indonesia. He graduated from the faculty of Engineering major in electrical engineering at Universitas Muhammadiyah Yogyakarta. His master and Ph. D degree has obtained from University Teknologi Malaysia major in Electrical and Electronic Engineering. Now he was a senior lecturer in Master of Electrical Engineering Department UNISSULA. He was published more than 100 academic papers in reputable conferences dan journal. His interest research in renewable energy, biomedical engineering, IoT, telematics and embedded technology.

Imam Much Ibnu Subroto is a member of Faculty of Industrial Technology, Universitas Islam Sultan Agung. He received the degree in Electrical Engineering majoring of Computer Engineering from Gadjah Mada University, Indonesia in 1998. He finished his study in Master of Computer Science from Universiti Teknologi Malaysia in 2007 and finished his Ph. D from Universiti Teknologi Malaysia majoring of Computer Science in 2015. Now he was a senior lecturer in Master of Electrical Engineering Department UNISSULA. He held of Information Technology Program in Master of Electrical Engineering Dept. He was published more than 100 academic papers in reputable conferences dan journals. His interest research in Data Mining, Business Intelligence, Big Data, Software Engineering, Smart Systems and Artificial Intelligence.