TELKOMNIKA, Vol.17, No.1, February 2019, pp.68~75
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v17i1.11627

Performance comparison of transmitting jumbo frame on Windows and Linux System

Supriyanto Praptodiyono*[1], Rian Sofhan[2], Anggoro S. Pramudyo[3], Teguh Firmansyah[4], Azlan Osman[5]
[1,2,3,4]Department of Electrical Engineering, Universitas Sultan Ageng Tirtayasa, Indonesia
[5]School of Computer Sciences, Universiti Sains Malaysia, Malaysia
*Corresponding author, e-mail: supriyanto@untirta.ac.id

Abstract

IPv6 is the successor of IPv4, the current Internet Protocol that runs out its address. It offers some improvements including simpler header format and extension header resulting in faster transmission of IP packets. However, IPv6 is a network layer protocol that requires lower layer services. IP packets from the network layer pass to data link layer to be encapsulated by layer 2 headers and trailer to become frames. Ethernet is the most widely used data link layer protocol in the current network devices. The technology is always improved to support high speed transmission. However, from standard Ethernet until ten gigabit Ethernet, the size of MTU remains unchanged at 1500 Bytes. This prevents the network from gaining an optimum performance on transmitting IP packets and operating systems cannot take full advantage of the high-speed performance of Gigabit Ethernet. This research aims to implement the transmission of IPv6 packets using jumbo frame on a test-bed environment. The implementation can be used to justify the impact of jumbo frame on the network as well as operating systems performance. The results prove that the OS used on implementation of jumbo frame affects on the network performance. The highest percentage of increasing throughput is 33.6% when both sender and receiver are running Windows. The decreasing delay by 54.36% was happened when using Linux in sender and Windows in receiver.

Keywords: *ethernet, ipv6 packet, jumbo frame, performance*

1. Introduction

The current Internet Protocol, IPv4, has been running out its IP address due to the rapid growth of Internet user. To overcome this problem, the new internet protocol had been developed so-called Internet Protocol version 6 (IPv6). IPv6 has a very large address spaces that is up to 128 bit [1]. The larger address space of IPv6 allows network to be extended for connecting every equipment which later is called as IoT (Internet of Things) [2, 3]. The IPv6 supports to send a large packet size up to 65 kB as indicated by the 16 bits next header field. In addition, it also allows to send jumbogram packet [4-6] that is larger than 65 kB size by introducing IPv6 extension header [4, 5]. However, the implementation of IPv6 should consider the technology used as its underlying layer since the conventional packets transmission has to pass the lower layer processing [7]. A number of underlying protocols is available today such as FDDI [8], Ethernet [9], and Frame Relay [10].

The current widely used of underlying technology is Ethernet that has gone to some evolutions starting from standard Ethernet to 10 Gigabit Ethernet [9, 11]. There are improvements on every generation including upgrading data rate and auto-negotiation. The most common type of Ethernet in use today is Fast Ethernet which is able to provide data rate up to 100 Mbps [12] and 1 Gbps Ethernet that can transmit packet with data rate up to 1000 Mbps. The frame of Fast Ethernet has maximum transmission unit (MTU) of 1500 Bytes as the maximum size of packet transmission. Even though the 1 Gbps Ethernet has an ability to transmit up to 1000 Mbps, its default MTU is still 1500 Bytes [7]. The MTU allows the sender machine to transmit packet up to 1500 Bytes only. In case it wants to send a large packet size, the packet should be fragmented into maximum 1500 Bytes per datagram. This will decrease the performance of packet transmission since it should transmit many datagrams to only send one file [13, 14]. In addition, this prevents operating systems from taking advantages of the

Received June 25, 2018; Revised November 8, 2018; Accepted December 9, 2018

Gigabit technology [13], [15-17]. The ability of the 1 Gbps Ethernet technology to send packet on high speed rate cannot be optimized [15]. Since the average size of files that are transmitted over the Internet has been far bigger todays [18, 19], the current size of MTU prevents network from gaining an optimum transmission speed and also burdens the CPU (Central Processing Unit) in processing packet data.

A number of previous research such as [16, 17], [20-25], tried to increase the MTU by introducing jumbo frame as well as super jumbo frame. Shaneel Narayan and Paula Raymond [16] implemented jumbo frame on a test-bed infrastructure with computers installed using Windows Server 2003/2008 operating system. Network and related metrics are measured for both network protocol IPv4 and IPv6. This research was conducted for transport protocol TCP and UDP. The results showed that jumbo frames produced throughput higher than normal frames for both TCP and UDP. Delay value of IPv6 was significantly lower than IPv4 for TCP and UDP packet dropped values were comparable between jumbo frame and normal frame. However, the implementation is only on Windows operating system and not on Linux system.

The author of [20] did a research on enhancing the network throughput by increasing Maximum Transmission Unit MTU. They find out that transmission of larger packets can improve the by reducing the IP packet overhead. The finding was supported by an analysis results using a mathematical model and simulation tools on wireless channels. The analysis was resulted in the decreasing of the performance of mathematical model when small MTU configuration is used. This is because the control traffic and installation times for small traffic is the same with the large traffic. However, it did not consider the performance of OS used in the experiments.

Supriyanto, Iznan H. Hasbullah and Rahmat Budiarto [23] conducted a research to evaluate the effect of IPv6 packet size on CEH (CRC Extension Header). CEH is an error control that is added to network layer while eliminating error control on the data link layer of intermediate nodes. The results showed that network latency decreased by 68% compared to normal latency as the packets size increased. The proposed error control was also capable of detecting errors while transmitting packet data as expected. Authors of [25] evaluated the internet performance when transmitting large MTU size. It revealed that the large MTU transmission offers much larger throughput. This experimental results were supported by Huston, G. [26] that theoretically calculated the increasing data efficiency by increasing the MTU size. However, the papers did not consider the performance of the OS used. Experiments on transmitting large packets size as well as large frame size on windows OS was conducted in [24]. Results of the experiment showed that larger packet is able to reduce the network latency as well as increases the throughput of the network. The usage of large frame size could reduce the number of frames transmitted. The experiments were done using UTP Cat 5e medium that has a very low bit error rate. The authors claimed that the usage of large frame could increase the network throughput up to 117% from the normal frame size. Since Linux system is usually used in networking machine rather than Windows, a research on evaluating its performance is required.

The transmission of jumbo frame can be done if the OS used both in sender and receiver support it. Since the OS is developed to support not only networking but also other functions, it need to know the ability of the OS on supporting the jumbo frame processing. Burjiz K. Soorty and Nurul I. Sarkar [27] compared the UDP packet of IPv4 and IPv6 with 4 modern operating systems (Windows 7, Windows Server 2008, Ubuntu 10.04, and Red Hat Enterprise Server 5.5). This research mainly wanted to know which OS was better in term of network performance. The results showed that: in term of throughput, IPv4 was better than IPv6 while Red Hat Server was better than Windows Server for both protocols. In term of RTT, IPv6 was better than IPv4 while Windows Server was better than Red Hat Server for IPv6. In term of jitter, IPv4 was better than IPv6 while Windows Server was better than Red Hat Server for both protocols. In term of CPU utilities, IPv6 was better than IPv4 while Ubuntu was better than Windows 7. This research did not exploit jumbo frame on the experiment and only uses normal frame.

Eric Gamess and Rina Suros [3] conducted a research by using an upper-bound model for measuring throughput of network protocol IPv4 and IPv6 on transmission protocol TCP and UDP. The models were established by all varieties of Ethernet (10, 100, and 1000 Mbps). The experiments were conducted using computers utilizing Windows XP SP2, Solaris 10, and Debian 3.1 as operating systems. This research found that the performance for both TCP and

UDP could reach the upper bound of Ethernet (10, 100, and 1000 Mbps). This research however did not utilize frame jumbo on the experiment.

This research aims to improve the network performance by increasing the MTU size in the form of jumbo frame transmission. The packet transmission was done using Windows OS and Linux as the widely used OS, in order to know the performance of the current available OS on processing the packet transmission. The rest of this paper describing the experimental network used in this research on section 2 that followed by discussing the results on section 3. The last section is a conclusion of this paper and potential future research.

2. Research Method

In order to evaluate the large IPv6 packets transmission using jumbo frame, an experimental network was set up. There are three nodes that represent a sender, a router and a receiver as shown in Figure 1. The entire node uses 1 Gbps Ethernet that allows the transmission of jumbo frames with file size of 5 MB. Four software are used in the experiment: two operating systems consist of Windows Server 2012 and Linux Ubuntu 14.04 LTS for clients, Windows 7 for PC Router, and Distributed Internet Traffic Generator (D-ITG) for measurement. D-ITG software is used to simulate Internet Traffic and accurately replicating appropriate stochastic processes for both IDT (Inter Departure Time) and PS (Packet Size) random variables over real IP networks [21]. There are three functions for measuring using D-ITG, those are: ITGSend for sending traffic, ITGRecv for receiving traffic, and ITGDec for decoding the logger file.

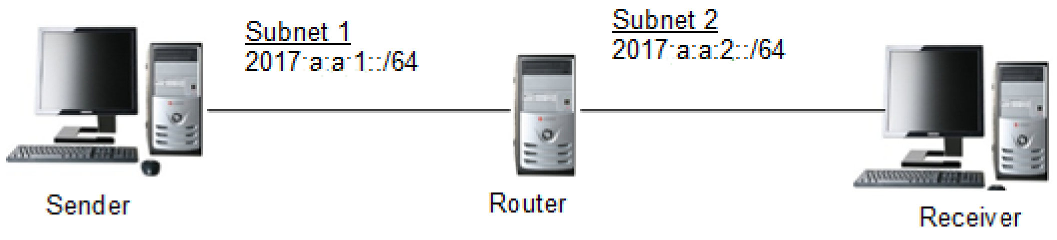

Figure 1. Experimental setup

Intermediate node in Figure 1 is used to represent the real network. As a router, it should be able to forward IPv6 packets. Sender host is set to send IPv6 traffics to the receiver host, using the ITGSend function. PC Router is purposed to give IPv6 address to each client so that sender host could be connected to receiver host. Receiver host is prepared to receive the IPv6 traffics from sender host, using the ITGRecv function. Finally, the ITG software logs the traffic in order to get the metrics data. There are four scenarios of operating systems to be evaluated. These scenarios are based on which operating system is installed on each client. The scenarios are presented as shown on Table 1.

Table 1. Scenarios of IPv6 Packets Transmission

Scenario	Sender	Receiver
1. (WS-WS)	Windows Server 2012 R2	Windows Server 2012 R2
2. (WS-LU)	Windows Server 2012 R2	Linux Ubuntu 14.04 LTS
3. (LU-WS)	Linux Ubuntu 14.04 LTS	Windows Server 2012 R2
4. (LU-LU)	Linux Ubuntu 14.04 LTS	Linux Ubuntu 14.04 LTS

3. Results and Discussion

Performance evaluation of IPv6 packets transmission was done by transmitting IPv6 packets using normal frame size (1500 bytes) and two jumbo frame sizes (4000 bytes and 9000 bytes). The transmission was conducted using 5 MB as the average multimedia file size [19]. Four network metrics are used in the measurement of network performance: round trip time, jitter, throughput, and CPU utility. The experiment was conducted with IPv6 as network protocol and TCP as the transmission protocol.

3.1. Round Trip Time

RTT (Round-trip time) defines the overall time needed by a bit or a packet to travel from a specific source to a specific destination and return back again to the sender. It includes propagation time, transmission time, queuing time and processing delay [7]. The propagation time depends on the medium used and the others are processed in the sender and receiver machine. This indicates that the performance of OS used is significant of the RTT calculation. The following results show RTTs from the experiment for 5 MB file size. Figure 2 is a graph that represents the average of RTT for 5 MB file size in 20 experiments. The vertical axis is the RTT that is measured in milliseconds and the horizontal axis is the MTU. The MTU represents the length of frame that carries IPv6 packets. As known, IPv6 performs fragmentation in the sender, so the 5 MB packet is fragmented based on the MTU to become smaller piece of packets before being sent. The RTTs are measured starting from the transmission of the first byte of IPv6 packets until an acknowledgment is received by the sender. Table 2 shows the average RTT and the percentage of decreasing RTT.

Table 2. RTT and Percentage of Decreasing RTTs

MTU	LU - LU		LU - WS		WS - LU		WS - WS	
Unit	ms	%	ms	%	ms	%	ms	%
1500	10.18	0	14.95	0	9.61	0	6.30	0
4000	9.63	5.41	12.99	13.10	5.45	43.25	3.18	49.47
9000	7.66	24.74	6.83	54.36	4.70	51.08	3.04	51.72

As a known, the RTT is measured when the whole packet is being transmitted through the network. The Table 2 shows that the usage of jumbo frame can decrease the RTT because the larger the frame size the number of frames will be lower. It demonstrates that the usage of jumbo frames can reduce the RTT significantly. The larger MTU is being used, the greater percentage of RTT decrease is obtained. The usage of MTU 4000 Bytes decreases the RTT with minimum percentage of 5.41% but yields 24.74% for MTU 9000 Bytes. For all scenarios, the MTU 9000 Bytes is able to decrease the RTT higher than the MTU 4000 Bytes especially if the sender is Linux Ubuntu, which indicates a significant increase in percentage. Figure 2 shows the trend of the decreasing RTT.

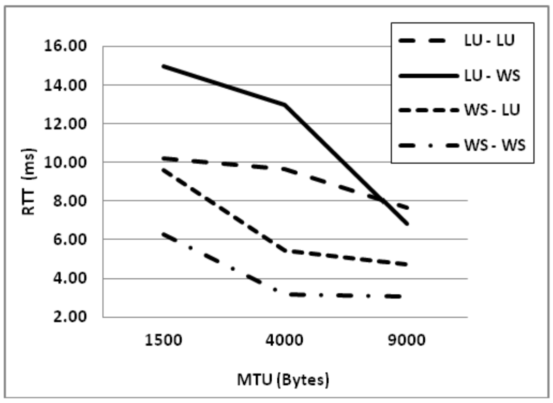

Figure 2. RTT of 5 MB file size transmission

As shown in Figure 2, the four scenarios show the same trend. In overall the trend is positive meaning the larger the frame, the lower the RTT. However, the scenario of WS–WS shows the lowest RTT for all MTU size. The minimum RTT in average is 3.04 ms in the transmission from WS–WS for MTU of 9000 Bytes. While the scenario of LU–WS indicates the most significant of decreasing RTT that reaches 51.72%. This can be logically interrelated with the number of fragmentations. For the file size of 5 MB and MTU 1500 Bytes, there are about 3333 frames instead of 1250 frames for MTU 4000 Bytes. When the MTU of 9000 Bytes is used, there are only 555 frames. Furthermore, this can be concluded that the usage of jumbo frame can decrease the RTT on all scenarios.

3.2. Jitter

Jitter is defined as the deviation of delay of every packet data that is transmitted over a network. It can be calculated by comparing the average delay of the packet transmission with an individual packet delay. It may a problem on a real time packet transmission. Table 3 shows the average jitter of the four scenarios. For all scenarios, the jitters are lower than 2 ms. However, the usage of jumbo frames is able to decrease the jitter. The percentage of the decreasing jitter is also shown in Table 3.

Table 3. Percentage of Decreasing Jitter

MTU	LU - LU		LU – WS		WS - LU		WS - WS	
Unit	ms	%	ms	%	ms	%	ms	%
1500	1.58	0	1.63	0	1.51	0	1.67	0
4000	1.29	18.26	1.42	13.37	1.17	22.13	1.60	3.97
9000	1.26	20.55	1.38	15.47	1.13	25.02	1.44	13.85

The frame with MTU of 9000 Bytes shows larger percentage of decreasing jitter when compared to the frames with MTU of 4000 Bytes. It means that the usage of MTU 9000 Bytes is able to increase network performance by decreasing jitter for both normal frames and with MTU of 4000 Bytes. By increasing of the size of frame, jitter could be reduced. From the four scenarios, the transmission of IPv6 packets from Windows Server to Linux Ubuntu provides the highest percentage. This means that congestion become minimum when Windows Server 2012 acts as sender and Linux Ubuntu acts as receiver.

Figure 3 shows trend of the jitter for the transmission of file 5 MB using four scenarios. The vertical axis is the jitter that is measured in millisecond and the horizontal axis is the MTU size. The usage of three MTU size is intended to obtain the influence of MTU to the network jitter. The local trend of all scenarios demonstrates the decreasing jitter from MTU 1500 Bytes to MTU 4000 Bytes until the MTU 9000 Bytes. However, in the scenario of Windows to Windows, it shows a different trend compared to the other scenarios. The decreasing jitter in this scenario especially from MTU 1500 Bytes to MTU 4000 Bytes shows its low decrease. As shown, the lowest jitter happens when Windows Server sends IPv6 packets to Linux Ubuntu. This means that congestion become minimum when Windows Server 2012 acts as sender and Linux Ubuntu acts as receiver.

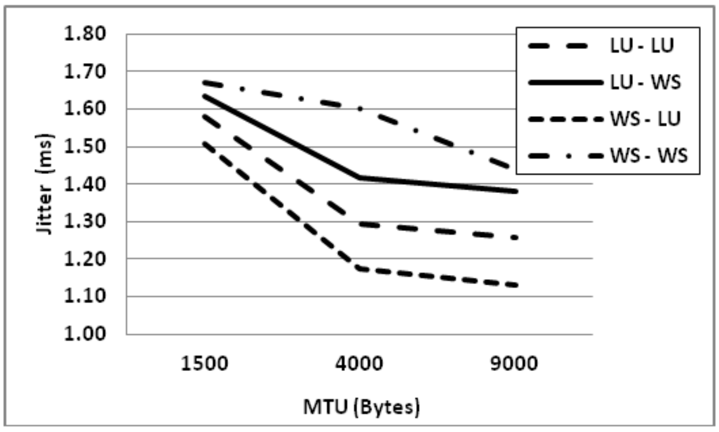

Figure 3. Jitter for 5 MB

3.3. Throughput

Throughput is defined as the transmission speed of packet data through a network. Throughput also refers to the actual bandwidth from a network link. Table 4 shows the experiment result of throughput for the transmission of 5 MB file size using the four different scenarios. It can be seen that for scenarios, the packet transmission using jumbo frame indicates a higher throughput. The Table 4 also shows the percentage of the increasing throughput for all scenarios.

Table 4. Percentage of Increasing Throughput

MTU	LU - LU		LU - WS		WS - LU		WS - WS	
Unit	Mbps	%	Mbps	%	Mbps	%	Mbps	%
1500	750.94	0	767.61	0	718.60	0	662.37	0
4000	895.53	19.25	894.50	16.53	899.73	25.21	767.29	15.84
9000	938.88	25.03	937.86	22.18	934.94	30.11	884.96	33.60

The percentage of increasing throughput of MTU 9000 Bytes is higher than MTU 4000 Bytes. This is because for file size 5 MB, there are only 555 number of frames instead of 1250 frames when transmitted using MTU 4000 Bytes. It means that increasing the jumbo frame size could reduce the overhead on IPv6 packets transmission. This is in line with the RTTs that decrease in the minimum value. The trend of the increasing throughput is shown in Figure 4.

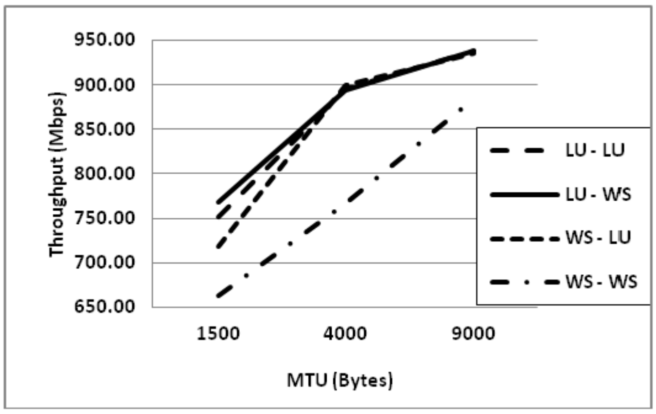

Figure 4. Throughputs for 5 MB

The vertical axis is the network throughput in Mbps and the horizontal axis is the MTU used to transmit IPv6 packets. In overall, the throughput increases for all scenarios. It shows similar trends except for the scenario of Windows to Windows that show a linear graph. The minimum throughput is 662.37 Mbps also occurs on Windows to Windows that uses MTU 1500 Bytes. However, in the scenario of Linux Ubuntu, it produces the highest average of throughput for every MTU. In contrary, in the scenario of Windows Server 2012, the scenario produces the lowest average of throughput for every MTU. This occurs because Windows Server 2012 reduces its RTTs to its minimum value and affects its throughput.

3.4. CPU Utilities

The usage of jumbo frame also influences CPU utilization. CPU utilization defines the cycles of processor that is used during transmission. Packets processing both in sender and receiver are based on bytes and thus the more bytes in the packets the more resources are needed. However, the more frames to be processed also require more resources. These experiments want to know the quantity of the number of frames versus the size of frames both in sender and receiver in different scenario.

This section discusses CPU utilization on the sender and receiver side due to the usage of jumbo frames that then is compared with the CPU utilization on processing the normal frame size. The experimental results were found that the using of jumbo frames are able to decrease the CPU utilization. Table 5 shows the percentage of decreasing CPU utilities from the experiment for the transmission of IPv6 packets of 5 MB size both for sender and receiver machine. It can be seen that the transmission using Windows Server utilizes very small CPU resources, there is no significant difference between MTUs. In contrast, Linux Ubuntu OS needs more CPU resources. However, the Linux Ubuntu shows the decreasing CPU utilities for jumbo frames. This means the usage of jumbo frame is able to decrease the CPU utilities on Linux Ubuntu especially in sender machine.

Table 5. Percentage of decreasing CPU utilities

MTU	Percentage of decreasing CPU utilities (%)							
	LU - LU		LU – WS		WS - LU		WS – WS	
	snd	rcv	snd	rcv	snd	rcv	snd	rcv
4000	6.96	11.05	2.63	0.00	5.28	6.23	0.00	14.29
9000	17.19	22.57	6.97	0.00	6.04	34.29	0.00	28.57

4. Conclusion

IPv6 is an ultimate solution of the IP address exhaustion that offers more benefits including IPv6 extension header. The option header includes facility to send jumbogram packet in order to reduce the number of packets sent as well as increasing throughput. However, Ethernet as the widely used underlying protocol still uses 1500 Bytes as its default MTU. This causes the optimally of the high data rate supported by the Gigabit Ethernet. A number of researchers tried to increase the size of MTU by sending IP packet in jumbo frame. Since the packet processing is conducted by the OS installed in sender and receiver, the performance of the common OS on processing jumbo frame need to be evaluated. The experiments were done by using jumbo frames on a network with TCP as the transmission protocol and IPv6 as the network protocol and also with modern operating system: Windows Server 2012 and Linux Ubuntu 14.04. From the analysis, we can conclude that the OS used on transmitting larger frame size could influence the network performance. In average, the usage of MTU 9000 Bytes, the throughput increased up to 33.60% on scenario WS-WS, latency decreased up to 54.36% on scenario LU-WS, jitter decreased up to 25.02% on scenario WS-LU, CPU utilities for sender and receiver both decreased by up to 17.19% on scenario LU-LU and 34.29% on scenario WS-LU. For future work, we are anticipating to include other operating systems from Linux distribution, Windows, and especially MAC OS. We are also anticipating evaluating other transmission protocols such as UDP and also other different scenarios such as evaluating application-level performance.

References

[1] Deering S and R Hinden, Internet Protocol Version 6 (IPv6) Specification, Internet Engineering Task Force, Request for Comments: 2460. December 1998.

[2] Soorty BK and N I Sarkar, UDP-IPv6 Performance in Peer-to-Peer Gigabit Ethernet using Modern Windows and Linux Systems. *International Journal of Computer and Information Technology*. 2014; 03(03): 496-502.

[3] Gamess E and H Ortiz-Zuazaga, Analytical Performance Evaluation of IPv6 and IPv4 Over 10 Gigabit Ethernet and InfiniBand using IPoIB. *International Journal Of Advanced Computer Science And Applications*. 2016; 7(8): 214-222.

[4] Hinden R, Internet protocol, version 6 (IPv6) specification. Internet Engineering Task Force, Request for Comments: 8200, 2017.

[5] Borman D, S Deering and R Hinden, IPv6 Jumbograms, Internet Engineering Task Force, Request for Comments: 2675, 1999.

[6] Scheelen Y and J Bosma, IPv6 Jumbograms, MSc Thesis, Amsterdam: System and Network Engineering University of Amsterdam, 2012.

[7] Behrouz A Forouzan, Data Communication and Networking 5th Edition, New York: McGraw-Hill, 2008.

[8] Ross FE, An overview of FDDI: The fiber distributed data interface. *IEEE Journal on Selected Areas in Communications*. 1989; 7(7): 1043-1051.

[9] Seno L, S Vitturi and C Zunino. *Real time ethernet networks evaluation using performance indicators*. Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation, 2009.

[10] Chase CJ, et al., Frame relay switched data service, US7257118B2 (Patents), 2007.

[11] Gallatin AJ, J S Chase, and K Yocum. *Trapeze/IP: TCP/IP at Near-Gigabit Speeds*. Proceedings of the USENIX Annual Technical Conference, 1999.

[12] Avallone S, Emma, Donato Pescap, A Ventre, Giorgioet. *A practical demonstration of network traffic generation*. Proceedings of the 8th IMSA. 2004.

[13] Winter R, R Hernandez and G Chawla. Ethernet Jumbo Frames, Ethernet Alliance, 2009.

[14] El Khadiri K, Labouidya O, Elkamoun N, Hilal R. Performance Evaluation of IPv4/IPv6 Transition Mechanisms for Real-Time Applications using OPNET Modeler. *Performance Evaluation*. 2018; 9(4).

[15] The Important of Jumbo Frames in Gigabit and 10-Gigabit Ethernet Networks, Small Tree Communications, White Paper. 2004

[16] Narayan S and P R Lutui. TCP/IP Jumbo Frames Network Performance Evaluation on A Test-bed Infrastructure. *International Journal Wireless and Microwave Technologies*. 2012; 6: 29-36.

[17] Iyer A P, Deshpande G, Rozner E, Bhartia A, Qiu L. *Fast resilient jumbo frames in wireless LANs*. Proceedings of the 17th International Workshop on Quality of Service, 2009.

[18] Sinha R, C Papadopoulos and J Heidemann. Internet packet size distributions: Some observations. USC/Information Sciences Institute, Tech. Rep. ISI-TR-2007-643, 2007.

[19] Angela. *Average File Sizes*. 2015; Available from: www.online-convert.com. [Accessed April 15, 2017].

[20] Guillén E, S Rodríguez and J Rodríguez, Throughput Optimization on Wireless Networks by Increasing the Maximum Transmission Unit. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*. 2013; 7(11):1400-1405.

[21] Salyers D, David Salyers, Yingxin Jiang, Aaron Striegel, Christian Poellabauer, JumboGen: dynamic jumbo frame generation for network performance scalability. *ACM SIGCOMM Computer Communication Review*. 2007; 37(5): 53-64.

[22] Rutherford W, Jorgenson L, M Siegert, P Van Epp, L Liu, 16 000–64 000 B pMTU experiments with simulation: The case for super jumbo frames at Supercomputing ’05. *Optical Switching and Networking.* 2007; 4: 121-130.

[23] Supriyanto, I H Hasbullah and R Budiarto, The effect of IPv6 packet size on implementation of CRC extension header. *Internetworking Indonesia Journal.* 2010; 2(2): 3-8.

[24] Supriyanto, Sofhan R, R Fahrizal and A Osman. *Performance evaluation of IPv6 jumbogram packets transmission using jumbo frames*. Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI). Yogyakarta, Indonesia. 2017.

[25] Murray D, Koziniec T, Lee K, Dixon M. *Large MTUs and internet performance*. Proceedings of IEEE 13th International Conference on High Performance Switching and Routing (HPSR). 2012.

[26] Huston G. *Mutterings on MTUs.* The ISP Column, 2009. Available from: http://wattle.apnic.net/papers/isoc/2009-02/mtu.pdf. [Accessed October 14, 2018].

[27] Soorty, BK and NI Sarkar. UDP-IPv6 Performance in Peer-to-Peer Gigabit Ethernet using Modern Windows and Linux Systems. *International Journal of Computer and Information Technology*. 2014; 03(03): 496-502.