Journal of Engineering and Applied Sciences 14 (4): 1084-1092, 2019
ISSN: 1816-949X

Analysis of the Performance of an IPv6 Network Through a Tunnel over the Double-Stack Mechanism End-to-End

Jorge E. Herrera Rubio
Department of EEST, Faculty of Engineering and Architectures, University of Pamplona, Bucaramanga, Pamplona, Colombia

Abstract: The present research deals with the study of one of the transition mechanisms from IPv4-IPv6 to determine the performance of the network with respect to the most important parameters such as delay and loss of packets over different bandwidths and storage buffer. Although, the process of migrating to native IPv6 has not been very fast, the study checks for improvements in IPv6 performance compared to IPv4. Results are presented on a pilot network using traffic measurements on the transport protocol in an empirical way.

Key words: Fragmentation, point-to-point, tunnel, migration, segmentation, measurements

INTRODUTION

In 2016 the IPv6 protocol completed 20 years of being active in the communications networks, its versatility for the handling that it has for the size of the header of 40 bytes with respect to the IPv4 protocol has allowed that the speed of the convergence (routing) and switching packets are much more efficient (Mercado *et al*., 2010).

The implementation of IPv6 is technically being carried out because the routing protocols such as BGP (Bolivar *et al*., 2012) are oriented to allow the connectivity between the different providers to the internet and are demanding the implementation by means of IPv6 prefix to improve the use of resources and infrastructure.

The most relevant features of the IPv6 protocol (Angel and Dominguez, 2014) among others are: simple implementation, allows end-to-end communication, no NAT service required, no broadcast, allows mobility, does not exist ARP, improves QoS, generation of addresses using EUI64 based address and router announcements, all IP addresses are public and is more secure.

The increase in new services, larger bandwidths, quality of services, mobility and the need for connectivity of more hardware devices has accelerated the growth of the internet to the point where the addressing capability for convergence made the Internet Engineering Task Force (IETF) (Hinden, 1991) will implement new addressing mechanisms so that, the transition between IPv4 and IPv6 is transparent for this reason the IPv6 protocol was created (Nordmark and Gilligan, 2005).

Literature review: There are research papers that show the differences in the performance of IPv4 and IPv6 protocols with respect to packet size (Shiwani *et al*., 2011), bandwidth, segment size, buffer size and unit size Maximum Transfer (MTU) in the switching and routing of packets.

The document RFC3142 (Hagino and Yamamoto, 2001) from the internet society network working group describes an IPv6-IPv4 Relay Translator (TRT) which allows IPv6-only hosts to exchange traffic (TCP, UDP) with the IPv4 hosts. A TRT system located in a centralized control can perform translation from {TCP, UDP}/IPv6 to {TCP, UDP}/IPv4 or vice versa.

Since, the size of the IPv6 header is twice that of IPv4 (Deering and Hinden, 1998), the performance is expected to decrease with respect to the average size of the global packet and the Maximum Transmission Unit (MTU) of the network topology physical layers, a situation that may be feasible, considering the amount of traffic and the processing capacity that can be handled in the hardware of the routers and computers active in the network.

Knowing that the MTU has a maximum of 1500 bytes in an Ethernet frame for transmission in the physical element any additional payload must fit within that MTU or in its default is divided into two or more full-size packages plus a package to complete the rest, that is you need more than one header.

By performing a theoretical analysis with the assignment of a 1460 byte MTU in all measurements made for IPv6 with 40-byte header (Shiwani *et al*., 2011), the percentage of the header occupies 2.73% of the total of the MTU to be transmitted when the data packet is small or larger than the MTU, a remaining small packet is produced which overloads the sending of packets and is reflected in the delay, thereof, examining for example, a packet of 44 bytes this will have 90.90% of the size occupied by its header.

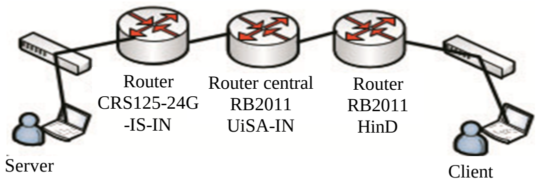

Fig. 1: Test scenario

MATERIALS AND METHODS

Experimental setup: The experiment is based on the configuration of three routers with hardware characteristics CRS125-24G-1S-IN (with 600 MHz CPU), RB2011UiAS-IN (with 600 MHz CPU) and RB2011HinD (with 1.4 GHz CPU) with router (OS) Operating System, configured in a controlled environment with a double-stack mechanism and 6-4 tunneling technique in order to allow the coexistence of the two IPv4 and IPv6 protocol simultaneously in order to differentiate the traffic in a Lan-Wan-Lan network as shown in the Fig. 1 with the establishment of an end-to-end communication.

The installed application for the generation and sending of packet traffic is JPerf Version 3.0.x which is adapted to the experiment by the UDP and TCP traffic handling capacity in the transport layer of the IP reference model. According to Shiwani *et al*. (2011) the application allows to measure the maximum performance of a network link, highlights the advantages such as: it is easy to use with GUI, less time needed to configure the process, the calculation of the bandwidth is automatic and the load and discharge of test is sequential or simultaneous.

Initially, the network is configure with fully static IPv4 addressing and the corresponding tests are performed generating UDP and TCP traffic by varying bandwidth, buffer size and packet size, then the sampling and finally the capture of the data; in the same way it is enabled for static addressing for IPv6.

The JPerf metric in TCP is: transfer time while in UDP are: transfer time, bandwidth, jitter and packet loss as they express it (Barayuga and Yu, 2015). Performance tests are used to validate the network using network segments with UTP cable and fiber connection cable to maintain uniform speed across the 100 Mbps network at the switch, router or network the client and server computers (Parsons and Griffith, 2015).

The idea of the research project is to allow coexistence and migration to IPv6 networks, using the dual stack transition mechanism to be able to perform the IPv6 configuration as an IP level solution (Nordmark and Gilligan, 2005) for this reason is divided in stages sequentially.

Active device configuration: Theend-to-end routing is set up and configure according to the proposed test scenary. The operating system of each of the host computers is configure with the IPv6 protocol, hardware and tunneling services on the physical interfaces. Physical IPv4 and IPv6 format tests are then run by using the PING command, both on the client and on the server, for example: ping -6 2002:68: c:214::2

Connectivity tests: In this stage, the configuration of the routers is made to establish the 6-4 tunnel as shown below: creation of the interface of the tunnel through which it will exit: interface 6-4 local-address = 10.1.0.25 mtu = 1280 name = 6-4 remote address = 172.26.1.1. The default path for the lan and wan interface is:/IPv6 route add distance = 1 dst-address = 2002:a0:19:1::1 gateway = SFP1 disabled = no distance = 1 scope = 30 target-scope = 10. b)/IPv6 route add distance = 1 dst-address = 2002:a01:19:2::1/64 gateway = ether 3 disabled = no scope = 0 scope = 10 target-scope = 10. The client of the tunnel interface is created by directing the tunnel: /IPv6 address add address = 2002:0a01:0019:1::1/3 advertise = no disabled = no eui-64 = no interface = 6-4. Add the IPv6 address of the LAN:/IPv6 address add address = 2002:0a01:0019:2::1/2 advertise = yes disabled = no eui-64 = no interface = ether 3.

Traffic generator software installation: According to research carried out by Mata (2015) traffic generators are classified into five categories according to the measures used in validation perspective: reproduction engines, generators maximum performance, used to test end-to-end network performances such as the JPerf that was used for bandwidth testing, jitter delay and loss rate as it is available on several platforms, model-based generators, high-level and self-configurable generators and special stage generators.

In order to perform statistical measurements and captures in real time, there is a difference between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), TCP is a reliable protocol (Pelaez, 2002), they reach the other end through the use of sequence and Acknowledgments (ACKs) and is oriented to the connection whereas with UDP the packets are sent without checking but with the advantage of being faster than TCP which makes JPerf (Mazalan *et al*., 2013) (Mazalan *et al*., 2013) take advantage of TCP and UDP capabilities to provide statistical data obtained in network links.

The configuration of the JPerf application on the client and server computer includes: the output format of the data in kbyte, listening port, UPD protocol (buffer size and packet size as shown in Fig. 2.

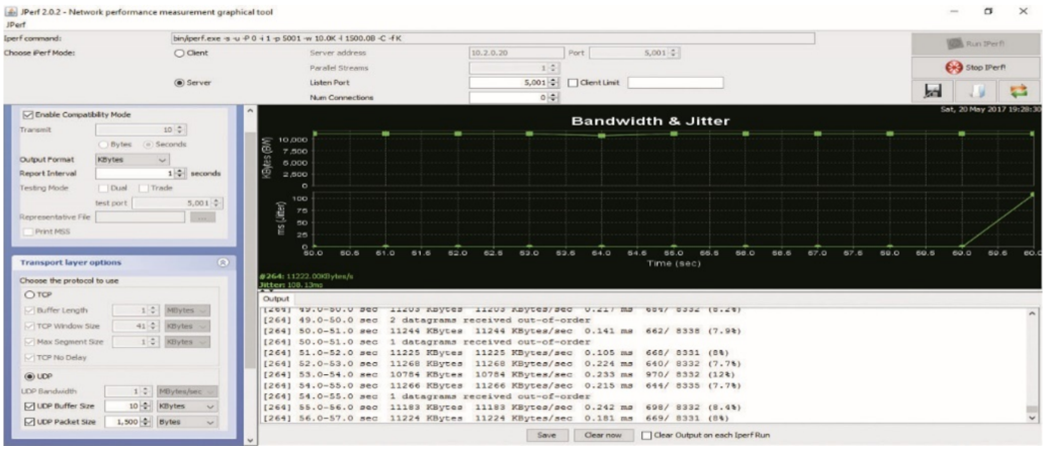

Fig. 2: Configuring UDP protocol parameters for IPv6

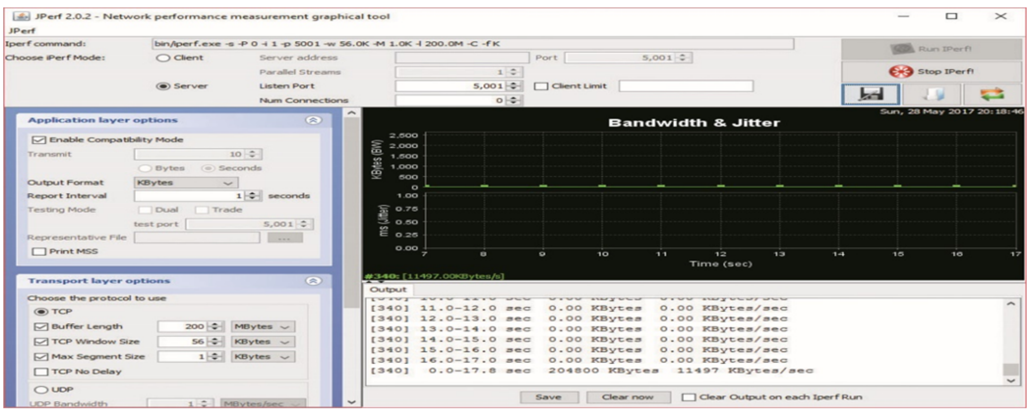

Fig. 3: Configuring TCP protocol parameters for IPv6

Then the TCP protocol is configure: the size of the buffer, the size of the TCP window and the maximum size of the segment, the IPv6 protocol (Fig. 3). The program JPerf places the server automatically in listening mode, the free port is selected (for example, the 5201 is taken and configure at both.

Capture and data processing: Real-time capture of the data is adjusted with 60 sec sampling for 10 tests for different bandwidths and different buffer sizes for UDP and for 60 sec for different buffer types by varying the size of the segments in the case of TCP, the captured data is stored in a flat text format file for the respective analysis, filtering and respective tabulation. With the use of Wireshark network protocol analyzer (Lamping *et al*., 2013) it is possible to perform permanent polling to verify that it sent TCP and UDP traffic by the channel is effective, some peaks in red represents loss of packets (Fig. 4).

TCP data capture and processing: Traffic generation starts for different buffer types of 10, 20, 30, 40, 50, 100 and 200 Mbyte by combining TCP segments of sizes 1 and 2 kbyte by means of different types of windows of 10, 20, 56 and 10 kbytes with random samples for 60 sec in order to avoid any inconsistencies in sampling. The log of the times in seconds were averaged which are evident according to the type of transport protoco (Kolahi *et al*., 2010) as seen in the example in Table 1 for the different TCPs Buffer in IPv4 and IPv6 protocols with 10 kbytes size window and 1 kyobyte segments.

It is also observed that the total bytes sent for a Maximum Segment Size (MSS) is maintained at an average proportion of the total bytes sent indicating that the IPv6 protocol limits sending based on the MSS regardless of the size of the buffer which is why the length of the window in the sending of the TCP segment is approximately one second while in IPv4 the total bytes sent if they increase in proportion to the size of the buffer but the time of sending is older.

In order to group the measurements for IPv6 (according to the evaluation methodology) with different buffer and varying the size of the TCP window but with a

Table 1: Time variation in seconds for different TCP buffer with IPv4 and IPv6 protocol for window from 10 kbytes to 1 kbyte of MSS

TCP buffer size (Mbyte) IPv4	TCP windows size kbytes	Max segment size (kbyte)	Total kbytes sent TCP 10 (kbytes) (IPv4)	Average transmitted kbytes (IPv4)	Bandwidth kbytes/sec (IPv6)	10 Kbyttes window (sec) (IPv4)	Total kbytes sent TCP 10 kbytes (IPv6)	Average transmited kbytes (IPv6)	Bandwidth kbytes/sec (IPv6)	10 kbytes (sec) window (IPv6)
10	10	1	614400	10240	11497	0.8900	665580	11093	11039	1.004
20	10	1	1228800	20480	10793	1.8970	655320	10922	10748	1.016
30	10	1	1843200	30720	10862	2.8280	675840	11264	10796	1.043
40	10	1	2457600	40960	10320	3.9690	655320	10922	10452	1.044
50	10	1	3072000	51200	10604	4.8280	665580	11093	10837	1.023
100	10	1	6144000	102400	10673	9.5940	635340	10589	9836	1.076
200	10	1	12288000	204800	11497	17.813	819180	13653	10704	1.275

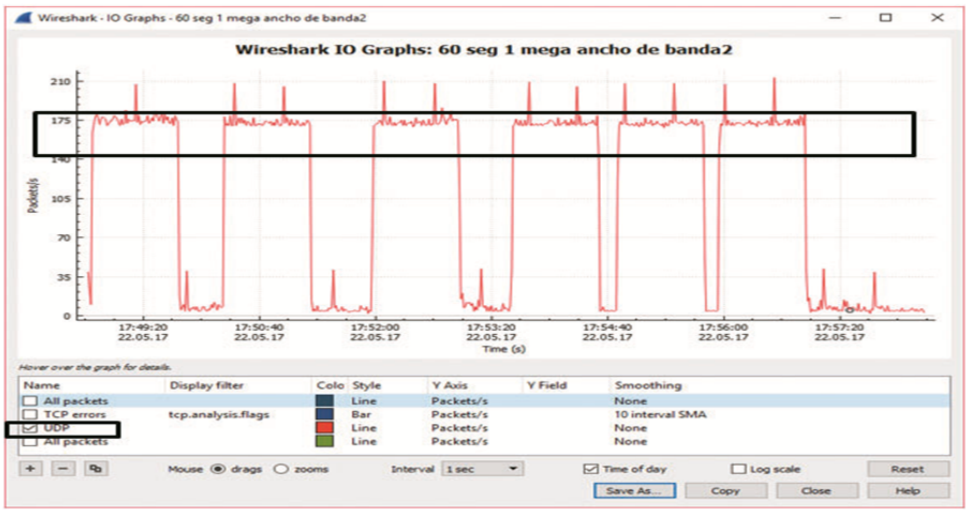

Fig. 4: Capture IPv6 datagram for 60 sec for 1 Mbps bandwidth

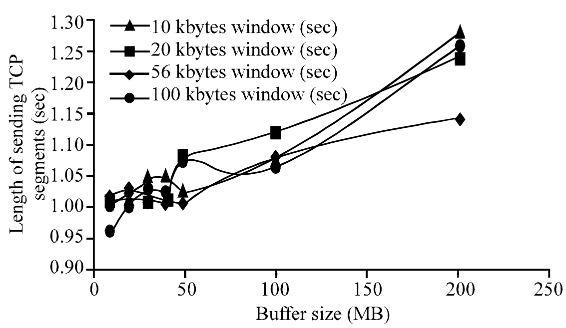

Fig. 5: Delays for IPv6 protocol with 1 kbyte segment

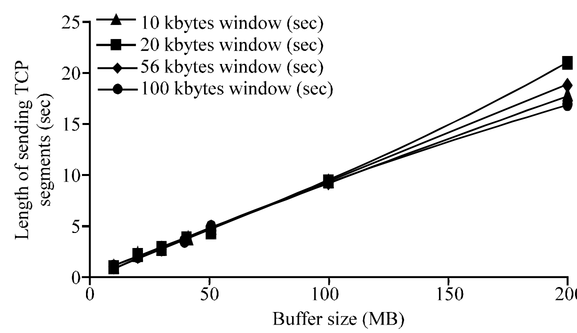

Fig. 6: Delays for IPv4 protocol with 1 kbyte segment

MSS of 1 kbyte, we can see how the sending times are maintained in approximately 1 sec (Fig. 5), thus, increasing the size of the buffer.

In the case of the sending of TCP IPv4 segments in the grouping of the measurements it is noticed that the sending times increase considerably as the buffer size increases which causes delays in the arrival times (Fig. 6) but no packet loss occurs.

Performing the analysis for a size of the MSS for 2 kbyte in IPv4, it is clear that the times increase much more than when analyzed for a MSS of 1 kbyte, (Table 2) column of "Total kbyte sent TCP to 10 kbyte".

To contrast the above and keeping the MSS data in 2 kbyte but in IPv6, it is perceived that the "Total kbyte sent TCP 10 kbytes" in Table 2 is reduced in a proportion of 45-50%, comparing with Table 1, there is shown that the sending times are below 1 sec, this is because the protocol to be tunneled keeps the times in order to avoid errors in the transfer of data and segment overlaps by the increase of MSS (Zhou *et al*., 2008).

When grouping the measurements for IPv6 with different buffer and varying the size of the TCP window

Table 2: Time variation in seconds for different TCP Buffer for IPv4 and IPv6 protocol with 10 kbytes window for 2 kbyte of MSS

TCP buffer size (Mbyte) IPv4	TCP windows size kbytes	Max segment size (kbyte)	Total kbytes sent TCP 10 kbytes (IPv4)	Average transmitted kbytes (IPv4)	Bandwidth kbytes/sec (IPv6)	10 Kbyttes window (sec) (IPv4)	Total kbytes sent TCP 10 kbytes (IPv6)	Average transmited kbytes (IPv6)	Bandwidth kbytes/sec (IPv6)	10 kbytes (sec) window (IPv6)
10	10	2	614400	10240	9752	1.0500	317400	5290	5705	0.927
20	10	2	1228800	20480	10771	1.9010	245760	4096	4455	0.919
30	10	2	1843200	30720	10030	3.0620	399360	6656	7051	0.943
40	10	2	2457600	40960	10239	4.0000	338460	5641	5895	0.956
50	10	2	3072000	51200	10590	4.8340	409680	6828	6923	0.986
100	10	2	6144000	102400	10699	9.5700	409560	6826	7098	0.961
200	10	2	12288000	204800	11354	18.037	374700	6245	6821	0.915

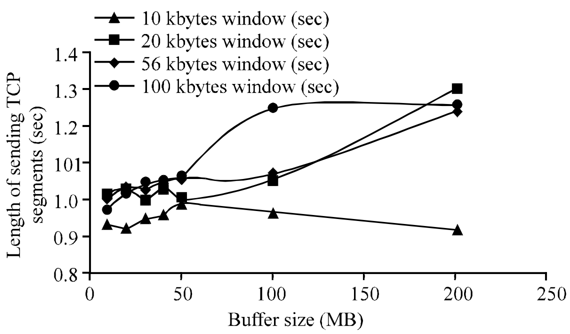

Fig. 7: Delay different windows IPv6 protocol with 2 kbyte segment

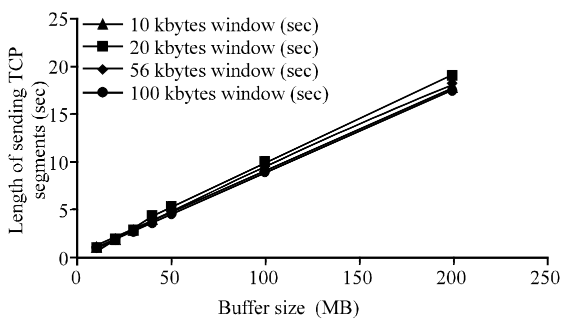

Fig. 8: Delay different windows IPv4 protocol with 2 kbyte segment

but with a MSS of 2 kbyte, we can see how times border the limits between 0.9 and 1.25 sec (Fig. 7) increase the size of the buffer.

In the case of the sending of TCP segments in IPv4 in the grouping of measurements it is noticed that the sending times increase between 1 and 19 sec in a large way as the size of the buffer increases with high delays as seen in Fig. 8.

This means that in the IPv4 protocol the sending of segments over the IPv6 protocol prevails, causing very poor performance to be present when segment size is changed which does not guarantee that an IPv6 tunnel is improved the sending of segments with respect to IPv4 because the process of fragmentation is different when the reception window size of the TCP buffer is very small, the receive window buffer is often invalid and the flow control mechanism interrupts the transfer until the receive buffer is empty (Mazalan *et al*., 2013).

UDP data capture and processing: In UDP, the parameterization of the datagram structure is executed by generating traffic with different types of 10, 64 and 100 kbyte buffer combining 1, 10 and 100 Mbps bandwidths for a 1460 byte UDP packet size Transfer (MTU) that restricts the size of the packets in the scheduling and sent them through the transport protocol (Mathis and Heffner, 2007), the samples are taken during 10 trials randomly every 60 sec, respectively as shown in Table 3.

In the sending of datagrams it is observed that independent of the UDP buffer that varies of 10, 64 and 100 kbytes, the size of the datagrams vary according to the Bandwidth (BW) in the case of Table 3 when this is 1 Mbps the average datagrams received are 5001 bytes and the average of each datagram is 84 bytes for an MTU of 1460 if the BW is 10 Mbps average received datagrams are 49994 bytes and the average of each datagram is 834 and finally for a BW of 100 Mbps the received data grams are 499994 bytes and the average of each datagram is 8834 bytes which demonstrates an approximate ratio of 10 times the speed to be able to do the distribution of the datagram over the channel width and ensure the sending time during the 60 sec for each sampling.

The loss rate of datagrams is zero when the UDP Buffer varies for a 1 Mbps BW which was not plotted, otherwise it is observed in Fig. 9 where the losses are high as the BW increases for 10 Mbps is approximately 2% and when BW is 100 Mbps is 70% as seen in variations of the UDP Buffer for 10, 64 and 100 kbytes.

This causes delays in the case of a 10 kbyte buffer is less significant but if the buffer is 64 kbyte the times are higher for 1 Mbps and 100 Mbps bandwidths (Fig. 10).

Likewise, if the buffer is increased to 100 kbyte (Fig. 11) the delaysare beyond of milliseconds, so, it is not advisable to increase the buffer when the standard (Postel, 1999) establishes limits where the width band,

Table 3: Parameterization of UDP datagrams for buffer size of 10 kbytes for IPv4

Events	UDP bandwidth Mbytes/sec (Mbps)	UDP buffer size (kbyte)	Received datagrams	Average size datagram	Lost kbytes	Loss of datagrams (%)	Lost datagrams	Bytes retained (kbyte)	Jitter (msec)
1	1	10	5001	84	0	0.00	5040	7326	0.450
2	1	10	5001	83	0	0.00	0	7326	0.303
3	1	10	5001	84	0	0.00	0	7326	0.522
4	1	10	5001	85	0	0.00	0	7326	0.508
5	1	10	5005	83	0	0.00	0	7326	0.486
6	1	10	5001	84	0	0.00	0	7326	0.399
7	1	10	5001	83	0	0.00	0	7326	0.377
8	1	10	5001	83	0	0.00	0	7326	0.446
9	1	10	5001	84	0	0.00	0	7326	0.395
10	1	10	5001	84	0	0.00	0	7326	0.524
1	10	10	49994	834	19	0.04	45	73206	0.486
2	10	10	49994	836	1	0.00	268	73232	0.394
3	10	10	49994	834	6	0.01	884	73225	0.377
4	10	10	49994	834	42	0.08	658	72345	0.446
5	10	10	49997	834	268	0.54	699	72845	0.337
6	10	10	49995	833	884	1.77	670	71925	0.395
7	10	10	49996	832	658	1.32	711	72272	0.301
8	10	10	49998	834	99	0.20	845	72655	0.366
9	10	10	49995	834	67	0.13	904	72331	0.433
10	10	10	49998	833	711	1.42	987	72198	0.315
1	100	10	499813	8333	303303	60.68	303303	278856	0.599
2	100	10	499945	8400	298011	59.61	298011	295802	0.519
3	100	10	499813	8399	303456	60.71	303456	287632	0.577
4	100	10	499952	8337	296184	59.24	296184	288488	0.516
5	100	10	499946	8335	306010	61.21	306010	284086	0.499
6	100	10	499995	8334	303649	60.73	303649	287543	0.546
7	100	10	499927	8379	311054	62.22	311054	276669	0.017
8	100	10	499993	8335	309345	61.87	309345	279182	1.199
9	100	10	499934	8334	310331	62.07	310331	277739	1.448
10	100	10	499929	8301	309584	61.93	309584	278826	1.162

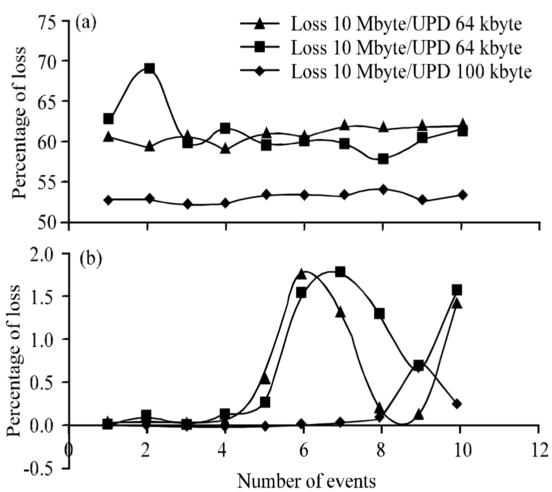

Fig. 9: a, b) Percentage of datagram losses according to bandwidth for IPv4

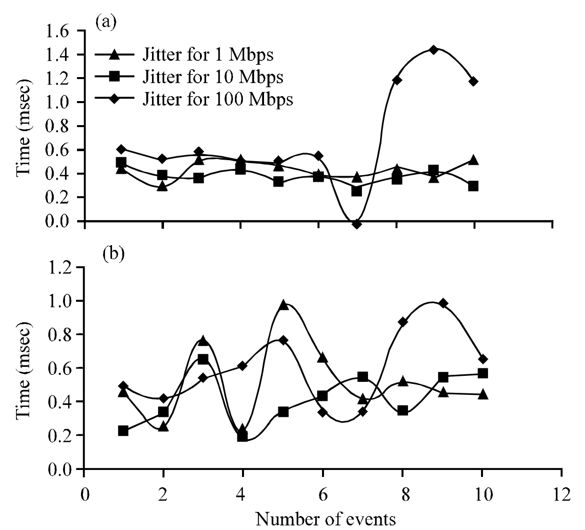

Fig. 10: a, b) Delay in milliseconds for 10 kbyte buffer and 64 kbyte for IPv4 UDP

MTU size and buffer size (Zhou *et al*., 2008), this information provides protection against badly routed datagrams within the header format.

In the case of IPv6 in Fig. 12 the percentage of losses is much lower than its counterpart in IPv4 for a BW of 10 Mbps this indicates that there is a better control in percentage of loss of packets but in BW of 100 Mbps the percentage of losses is between 50 and 75% similar to the analysis with IPv4, so, it is not recommended to increase the BW when working with 6 to 4 tunnel.

Examining for IPv6, Fig. 13 shows very low delays compared to what happens in IPv4 for each buffer because of IPv6 has a lower MTU limit of 1280 bytes. That

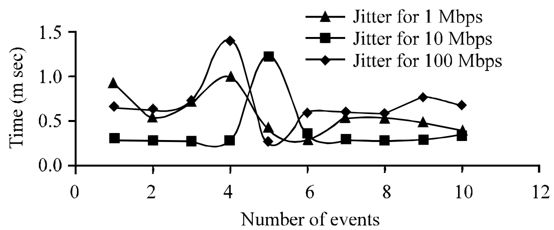

Fig. 11: Delay in milliseconds for 10 kbyte buffer and 64 kbyte for IPv4 UDP

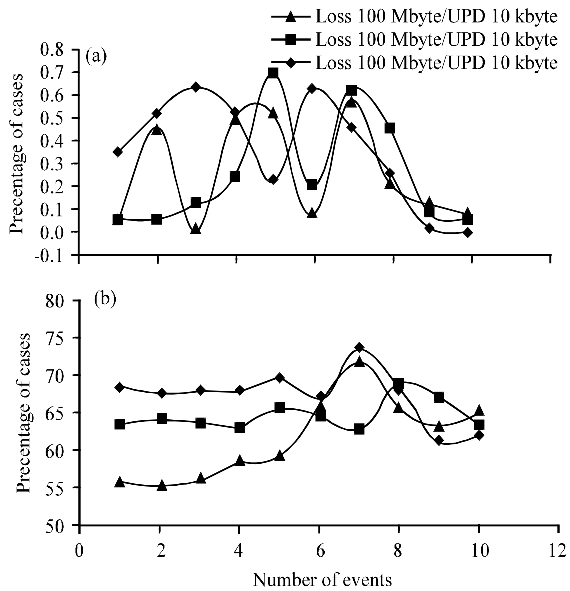

Fig. 12: a, b) Porcentaje de perdidas de datagrama segun el ancho de banda para IPv6

is IPv6 does not fragment packets below this limit. To send IPv6 packets over a link with an MTU of >1280 bytes (Anonymous, 2017), the link layer must fragment and transparently defragment IPv6 packets, so that, they can be processed over a tunnel over IPv4 in the case of IPv4, the minimum MTU is 576 byte.

Examining the delays in Fig. 13, the datagrams vary between 0.1 and 0.8 msec, thus, changing the sizes of the buffers, this indicates an improvement in the control of the sending of the datagrams decreasing the loss of them.

RESULTS AND DISCUSSION

The test network performance analysis focused on the generation of traffic to verify the operation of the TCP and UDP transport protocol over IPv4 and IPv6, respectively. Table 1 shows the buffer variation for several TCP Buffer values in Mbyte when the window size

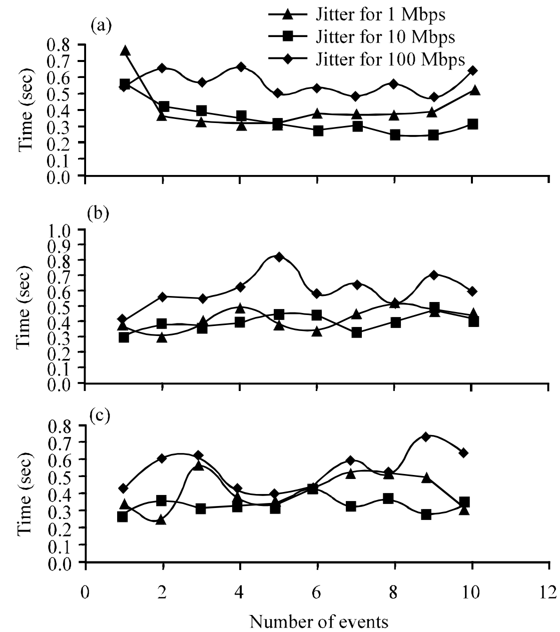

Fig. 13: a-c) Delay in milliseconds for 10 kbyte, 64 kbyte and 100 kbyte Buffer for IPv6 UDP

is 10 kbyte with a MSS of 1 kbyte, the network performance in IPv4 is reduced because there are very long delays high when compared to IPv6 delays do not exceed the time of 1 sec and the sending of TCP packets retains an average of the data to send, therefore there is no fragmentation of the packets.

Now, if the 2 kbyte MSS is compared in Table 2, the delay times increase in IPv4 while in IPv6 they are below the second and there is a TCP packet sending that is reduced to half of what it had when the MSS was 1 Kbyte and as a consequence the bandwidth is reduced by half. In order to have a more complete reference, the window sizes were changed in the order of 20, 56 and 100 Kbytes (Fig. 5-8) and the very high changes in packet time delay in the use of the IPv4 protocol and minimum in the IPv6 protocol.

When the MTUs in TCP are increased, the Bit Rate Error (BER) increases (Rodriguez, 2012), a situation that can be improved by using IPv6 end-to-end links by activating extension headers such as "Hop-to-Hop" (Anonymous, 2006) as long as the routers have this function active, there are also processing requirements in several network devices for the case study the tunnel system used is with "Next Header" header that by default is TCP (Kent, 2005), so, there is a control via the double-stack mechanism for IPv6 functionality. In this scheme, the routers do not perform the fragmentation process reducing the time for switching and sending of the packets.

CONCLUSION

When examining the transfer time metrics in TCP, it is observed an increase in the delay of the sending of the packages when making two jumps to the extent that it increases the size of the buffer there is an improvement in the sending of TCP segments over a bandwidth of 10 Mbps in the case of IPv4, the opposite situation happens in IPv6 where the transfer times are maintained in a second and the TCP segments maintains an average size over a bandwidth of 5 Mbps because it is not necessary the fragmentation of the segments, the fragmentation is performed at the source prior to the discovery of the smallest MTU through the path MTU discovery process.

In IPv6 the percentage of variation is reduced by improving the problem of delay and fluctuation. In IPv4 the fragmentation process generates more delay and fluctuations by use of flow control between different routers and active computers. Packet delay is also affected by the amount of traffic being carried on the network and the processing capacity of the active equipment in the network because if the delay in very large packet loss rate on the routes fluctuate in a proportion very high. As the bandwidth is increased for sending datagrams, the size of the datagrams get bigger by a ratio of 10 but at the same time the loss of the datagrams enlarge and the variation of the jitter is high.

In the case of UDP metrics, the percentage of datagram losses for bandwidths up to 10 Mbps is close to 1.8% and for 100 Mbps of 65%, if the buffer for the different bandwidths (1, 10 and 100 Mbps), the delays converge to 1.4 msec in the case of IPv4, in the case of IPv6 the loss percentages for 10 Mbps are 0.7% and for 100 Mbps of 75% when the respective buffer is varied, the delays are up to 0.8 msec, it is here that a time improvement advantage of 57.14% is observed.

When an end-to-end connection is established, the smallest MSS values will be used for information exchange in order to avoid fragmentation in the IP layer but because there is an intermediate network with routers it is possible that TCP packets are fragmented due to to that the size of the MSS changes in the process of encapsulating the layers.

Using too large a buffer generates a delay in the network because the buffer filling must be completed in the MTU conformation process which affects the increase of the error rate, packet fragmentation and a complex control of the TCP flow.

The dual-stack mechanism is the most used in the transition processes which guarantees the coexistence of IPv6 without affecting IPv4 (Alayon *et al*., 2015) and work transparently, so that, end users the transition with the help of the 6-4 tunnel is performed as a point-to-point unicast link layer model to allow the encapsulation of IPv6 packets.

REFERENCES

Alayon, C.A.M., R.F. Escobar and V.J.A. Zambrano, 2015. Implementation of transition and coexistence mechanisms for IPV4-IPV6 protocols in computer centers on supported high performance academic networks. Sistemas Telematica, 13: 83-106.

Angel, F.M. and J.F.C. Dominguez, 2014. Implementation of Ipv6 services at the Autonomous University of Guerrero, Mexico. Vinculos, 10: 393-400.

Anonymous, 2006. IPv6 extension headers review and considerations. Cisco Systems, San Jose, California, USA.

Anonymous, 2017. IBM documentation 7.2. IBM, Armonk, New York, USA. https://www.ibm.com/support/knowledgecenter/es/ssw_ibm_i_72/rzahg/ic-homepage.htm

Barayuga, V.J.D. and W.E.S. Yu, 2015. Packet level TCP performance of NAT44, NAT64 and IPv6 using iperf in the context of IPv6 migration. Proceedings of the 5th International Conference on IT Convergence and Security (ICITCS'15), August 24-27, 2015, IEEE, Kuala Lumpur, Malaysia, ISBN:978-1-4673-6537-6, pp: 1-3.

Bolivar, L., F. Guerrero and O. Polanco, 2012. [Design and implementation of an IPv6 network for efficient transition from IPv4 (In Spanish)]. Ing. Compet., 14: 179-189.

Deering, S.C. and R.N. Hinden, 1998. Internet protocol, version 6 (IPv6) specification. Stand. Track, 6: 1-39.

Hagino, J. and K. Yamamoto, 2001. An IPv6-to-IPv4 Transport relay translator status. Internet Soc., 1: 1-11.

Hinden, R., 1991. Internet protocol, DARPA internet program. Defense Advanced Research Projects Agency, Arlington County, Virginia.

Kent, S., 2005. IP authentication header. Internet Soc., 1: 1-34.

Kolahi, S.S., N. Chand, B.K. Soorty and Z. Qu, 2010. Performance comparison of IPv4 and IPv6 in peer-peer and client server local area networks. Proceedings of the IEEE 10th International Conference on Computer and Information Technology (CIT'10), June 29-July 1, 2010, IEEE, Bradford, England, UK., ISBN:978-1-4244-7547-6, pp: 41-45.

Lamping, U., R. Sharpe and E. Warnicke, 2013. Wireshark user â€TM s guide. Free Software Foundation, Boston, Massachusetts.

Mata, E.R., 2015. [Evaluation of tools for traffic generation scenarios]. Master Thesis, Technical University of Madrid, Madrid, Spain. (In Spanish)

Mathis, M. and J. Heffner, 2007. Packetization layer path MTU discovery. Internet Eng. Task Force, 1: 1-32.

Mazalan, L., S.S.S. Hamdan, N. Masudi, H. Hashim and R.A. Rahman *et al*., 2013. Throughput analysis of LAN and WAN network based on socket buffer length using JPerf. Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE'13), November 29-December 1, 2013, IEEE, Mindeb, Malaysia, ISBN:978-1-4799-1508-8, pp: 621-625.

Mercado, G., C. Taffernaberry and A.D. Piccolella, 2010. Design and simulation of the implementation of IPv6 protocol transition technologies and procedures in INTRANETS using an IPv6 test bed. Master Thesis, Universidad de Mendoza, Mendoza, Argentina.

Nordmark, E. and R. Gilligan, 2005. Basic transition mechanisms for IPv6 hosts and routers. Internet Eng. Task Force, 1: 1-27.

Parsons, K.R. and J. Griffith, 2015. How to guide on JPerf and IPerf. ChaseBros, Perry, Utah.

Pelaez, R.S., 2002. Security Analysis of the TCP/IP Protocol Family and its Associated Services. O'Reilly Media, Sebastopol, California, USA.,.

Postel, J., 1999. User datagram protocol. Internet Eng. Task Force, 1: 1-3.

Rodriguez, I., 2012. Link-level performance analysis on next-generation Wi-Fi and gigabit ethernet channels for ipv6 jumbograms. Grupo De Investigaci_on En Seguridad Y Sistemas De Comunicaciones (GISSIC), Bogota, Colombia.

Shiwani, S., G.N. Purohit and N. Hemrajani, 2011. Performance analysis of IPv4 v/s IPv6 in virtual environment using UBUNTU. Intl. J. Comput. Appl., 1: 72-76.

Zhou, X., M. Jacobsson, H. Uijterwaal and P.V. Mieghem, 2008. IPv6 delay and loss performance evolution. Intl. J. Commun. Syst., 21: 643-663.