computers

MDPI

Article

Non-Fragmented Network Flow Design Analysis: Comparison IPv4 with IPv6 Using Path MTU Discovery

Thiago Lucas [1], Maycon Ferreira [2], Rychard Plachta [2], Gabriel Ferreira [1,*] and Kelton Costa [1]

[1] Faculty of Sciences, Paulista State University Júlio de Mesquita Filho, São Paulo 17033-360, Brazil; t.lucas@unesp.br (T.L.); kelton.costa@unesp.br (K.C.)
[2] Information Security Department, Ourinhos Faculty of Technology, São Paulo 19910-206, Brazil; maycon.ferreira@fatec.sp.gov.br (M.F.); rychard.plachta@fatec.sp.gov.br (R.P.)
* Correspondence: vieira.ferreira@unesp.br

Received: 26 May 2020; Accepted: 18 June 2020; Published: 26 June 2020

Abstract: With the expansion in the number of devices that connect to the Internet, a new area, known as the Internet of Things (IoT), appears. It was necessary to migrate the IPv4 protocol by the IPv6 protocol, due to the scarcity of IPv4 addresses. One of the advances of IPv6 concerning its predecessor was the Path MTU Discovery protocol, in which this work aims to demonstrate its effectiveness in a virtual environment. Employing the VirtualBox virtualization program, a scenario is defined with fifteen machines with the Debian operating system and two network scenarios, one using the IPv4 network configuration and the other using the IPv6 network configuration. In both situations, the MTU values of all machines were changed to perform the performance tests while using UDP transport traffic. The fragmentation of packets demonstrated the effectiveness of the Path MTU Discovery protocol. The results achieved point to a stabilization in bandwidth and jitter when Path MTU Discovery is used and to change when it is no longer applied, proving its effectiveness.

Keywords: MTU; IPv4; IPv6

1. Introduction

According to Comer [1], the Global Internet started, when the Advanced Research and Projects Agency (ARPA) began to adopt the TCP/IP protocol on computers that were connected to its network. However, in Internet Protocol version 4 (IPv4), the increase in the number of people connected to the Internet was not expected, causing their addresses to be increasingly close to being exhausted. Some measures were created to delay this exhaustion, such as Network Address Translation (NAT), Dynamic Host Configuration Protocol (DHCP), and Virtual Host.

To the Internet, there are two types of addressing, IPv4 and IPv6. According to Torres [2], they have the main function of sending each computer or device connected to the Internet, where there are no same addresses. In 2010, IPv4 was used by 99% of computers globally. In surveys that were conducted by GoogleIPv6 [3] of users who access Google, 29.01% use the IPv6 address.

IPv6 also adds other new features and improvements over IPv4, such as simplification of the header, extension header, new address format, and autoconfiguration support.

Other points have been changed, the fragmentation and reassembly of packets with fragmentation only at the source, IPv6 uses the Path MTU Discovery protocol, which describes the Maximum Transmission Unit (MTU), the maximum packet size that can travel on the network and seeks to guarantee the sending in the broadest possible extent, unlike IPv4, which can fragment more than once to adapt the packet to the MTU size.

The present work seeks to present the effectiveness of the IPv6 Path MTU Discovery protocol. We tested the efficiency with the traffic performance obtained in similar tests in IPv4 networks that show fragmentation scenarios. The lack of fragmentation due to the action of Path MTU Discovery (fragmented IPv6 scene) makes the jitter and bandwidth values stable, with or without fragmentation, which proves the effectiveness of MTU discovery when along the way. It reinforces the situation of fragmentation while using IPv4 where there is no Path MTU Discovery implement, and the fall in bandwidth, as well as the increase in jitter, becomes clear. In short, Path MTU Discovery allows for the quality of traffic to remain the same, regardless of the presence of routers that do fragmentation. It is also worth mentioning as another contribution that this subject does not have many related studies, being, therefore, an essential work in this context.

It is worth mentioning that the contribution of this work has two important aspects: using Path MTU Discovery, it is possible to maintain the levels of jitter and bandwidth and, in scenarios where Path MTU Discovery is not used, there is a significant drop in bandwidth and a considerable increase in jitter. The work proves that, when Path MTU Discovery is used, there is stability in the network observing the values of jitter and bandwidth.

Section 2 presents related works. Section 3 presents the materials and methods used to perform the tests in the laboratory. Section 4 illustrates the obtained results, and Section 5, finally presents the conclusion.

2. Related Work

Before the following text, it is worth noting that the present work, as compared to the related works, occurs in an unprecedented way, although they did not carry out an analysis of the effectiveness of Path MTU Discovery comparing IPv4 with IPv6; somehow, they performed other types of analysis of performance for applying or not the newest protocol.

Luckie and Stasiewicz [4] studied the behavior of Path MTU Discovery (PMTUD) on 50,000 popular websites and concluded that the error rate in IPv6 is enough as compared to IPv4. However, there are standard errors in the behavior of software that should be filtered by firewalls. If these are corrected, PMTUD failures are reduced by 63%. TCP Behavior Inference Tool was used to measure its tests. Narayan et al. [5], in their work, analyzed the performance of the IPv4 and IPv6 protocols in different Windows operating systems, while using TCP and UDP traffic between two nodes and the transfer rate as the metric, in their results they proved that the performance does not depend only on the chosen protocol, but also the operating system. For large packet sizes, the difference in IPv4 and IPv6 performance is 1.3% higher than the theoretical value. Narayan, Shang, and Fan [6] obtained similar results. In their work, they used six Windows and Linux operating systems that were configured with IPv4 and IPv6 and evaluated empirically for the difference in performance, used metrics, such as throughput, delay, instability, and CPU usage. There was a significant difference in the average delay between operating systems, with Windows having less latency by up to 5% and higher processing usage by up to 35% compared to Linux. Li et al. [7] presented a measurement study on the resources and traffic behavior of users over IPv4 and IPv6 on the network of the Xiam Jiaotong University campus, the differences in terms of the distribution of the average packet size, flow size, flow duration, and self were analyzed for similarity of packet size and traffic volume. The main result obtained was since users' IPv6 networks are more stable. Ali [8] demonstrated the two tunnels IPv4 and IPv6 and when migrating from IPv4 to IPV6 and the present risks.

The following works found that not all situations exist where IPv6 stands out from IPv4. Cabellos-Aparicio et al. [9] make a comparison that is based on the measurement of transfer for different mobile protocols, including mobile IPv4 and mobile IPv6; its results were severe fluctuations in the quality of service before the transfer to Mobile IPv4 and Mobile IPv6, the latter with unacceptable latency of transmission to real-time traffic. Mobile IPv4 performs better than mobile IPv6, because it does not need to run the Duplicate Address Detection and Neighbor Inaccessibility Detection algorithms. Yasinovskyy et al. [10] compared VoIP performance on IPv6 and IPv4 LANs in the

presence of varying levels of UDP traffic in the background, while using maximum and average jitter, packet loss, Mean Opinion Score, and throughput as metrics. The maximum jitter for IPv6 was slightly higher than for IPv4, and packet loss can reach almost 18% for IPv4 and 24% for IPv6, and MOS decreases significantly. Hyun et al. [11] focused on evaluating the performance of IPv6-based mobile networks, analyzed detailed TCP connection procedures according to the different versions of IP on the hosts. They also compared the performance difference between IPv6 and IPv4 by measuring the time to establish a connection with 40 popular sites. Once again, IPv4 was better than IPv6, due to the immaturity of the IPv6 infrastructure.

Luckie, Cho, and Owens [12], a review of the faults present in the PMTUD are carried out, a tool designed to help infer the location of a fault was also presented, providing more details about the faults. Göhring, Shulman, and Waidner [13] explored the benefits versus disadvantages of PMTUD on the internet from the client/server perspective. They showed that PMTUD exposed customers to the degradation of attacks in ICMP mechanisms and proposed a countermeasure for this. Xiao et al. [14] presented a model, called PF-Fragment Packet in Gateway (PFPG), which can ensure that reassembly does not occur during packet forwarding, thus reducing the formation of errors.

Custura et al. [15] explored the size of the MTU path experienced on major Internet networks, wired, and mobile networks. They found that the Maximum Segment Size (MSS) field of the Transmission Control Protocol (TCP) protocol was used in 20% of the tested networks, in IPv6 servers they found that the MSS was hardly used, in addition, they detected black holes implemented by more than one-quarter of IPv6 web servers and almost 15% of IPv4 web servers. Supriyanto et al. [16] evaluated the performance of the transmission of IPv6 packets using jumbo frames, using IPv6 packets that were more significant than 1500 bytes in Windows operating systems. The results show that transmitting larger packets size using jumbo frame can increase the network throughput by up to 117%. In a similar work, Praptodiyono et al. [17] using jumbo in other test environments proved that the operating system used affects the performance of the network, with the highest percentage increase in transfer rate being 33.6% when the sender and the recipient are running Windows. Additionally, a decreasing delay of 54.36% occurred when using Linux at the sender and Windows at the receiver.

3. Materials and Methods

The present work had the deductive method concerning the research, using principles recognized as valid and indisputable.

Regarding the approach, quantitative research, with results that could be quantified, with analysis in numerical data employing statistical procedures, as to nature, and applied research, which sought to aggregate knowledge in practice and solutions to specific problems, in terms of objectives.

The critical analysis sought to identify the factors that implied the occurrence of the issues, justifying through results. In terms of procedures, an approach adopted experimental research for testing and data collection in a virtualized environment.

We also used searches in articles, books, and websites in the area of computer networks for bibliographic review. This analyzed what was obtained and developed the results and discussions to conclude the research.

A virtual environment is created using the Oracle VM VirtualBox software, the scenario described in Figure 1, with fifteen machines using the Debian 8.7.1 operating system. Each computer is also assuming the role of the router, having two network adapters that always connected to a different IPv4 network, except the first and the last. Each network adapter had different MTU values according to the proposed scenario.

Additionally the IPv4 scenario with standard MTU and fragmented MTU was used. Similarly, the IPv6 situation with standard MTU and fragmented MTU, to carry out the tests with the Iperf program, made it possible to analyze the network through the results obtained in the proposed scenarios.

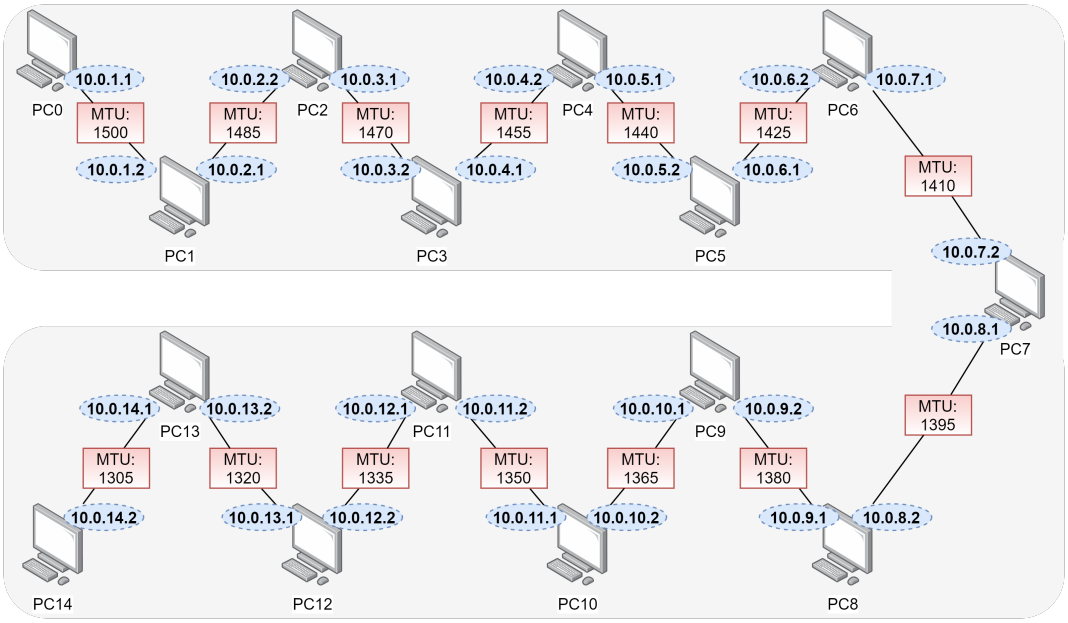

Figure 1. Proposed IPV4 Scenario. Source: Elaborated by the Authors.

3.1. Configuring the IPv4 Scenario

All of the devices on the network are connected to an internal network, where PC0 is connected to an adapter, one on the Internal Network sw1, with IP 10.0.1.1 configuration, default MTU 1500, and default gateway 10.0.1.2.

PC1, for once, is connected with the adapter on the Internal Network sw1, the adapter two on the Internal Network sw2, IP 10.0.1.2 configuration on eth0, the standard MTU of 1500, and IP 10.0.2.1 on eth1 with MTU of 1485.

Enabling packet routing on the machine by uncommenting the line: net.ipv4.ip_forward = 1 in the /etc/sysctl.conf file.

Adding the following commands in /etc/rc.local, to change the MTU value (ifconfig eth1 MTU 1485), the command to add a route to the routing table (IP route add 10.0.14.2 via 10.0.2.2) and command to enable NAT (iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE/iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE) to allow rewriting the source IP addresses of a packet that passed over a machine router that a computer on an internal network had access to an external system.

The configurations of the other machines followed that of PC1 described above, changing the Internal Networks of the adapters, the IP, and MTU values by their respective Figure 1. Adding just one more command in the /etc/rc.local to add one more route to the routing table (IP route add 10.0.1.1 via 10.0.2.1) to establish a connection from both PC0 to PC14 and PC14 to PC0.

The configuration arrived at PC13 with the same standards until a certain time. PC13 is connected with adapter one on the Internal Network sw13 and adapter two on the Internal Network sw14. Having a configuration of IP 10.0.13.2 on eth0 and MTU of 1320, and IP 10.0.14.1 on eth1 with MTU of 1305. Enabling packet routing on the machine, uncommenting the line: net.ipv4.ip_forward = 1 in the /etc/file sysct1.conf. And the commands in /etc/rc.local, for changing the MTU value (ifconfig eth0 MTU 1320/ifconfig eth1 MTU 1305), and adding just one route to the routing table with the command (IP route add 10.0.1.1 via 10.0.13.1) and command to activate NAT (iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE/iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE).

In PC14, the configuration was similar to that of PC0, where P14 was connected with adapter1 on the sw14 Internal Network. IP configuration 10.0.14.2, MTU 1305, and with default gateway 10.0.14.1. Add the command in /etc/rc.local, to change the MTU value (ifconfig eth0 MTU 1305).

Once these settings were made, PC0 had the connection established with PC14 through each machine, as contained in the topology of Figure 1.

3.2. Configuring the IPv6 Scenario

To the IPv6 scenario, the settings were made with the new IPs, according to the situation that is described in Figure 2.

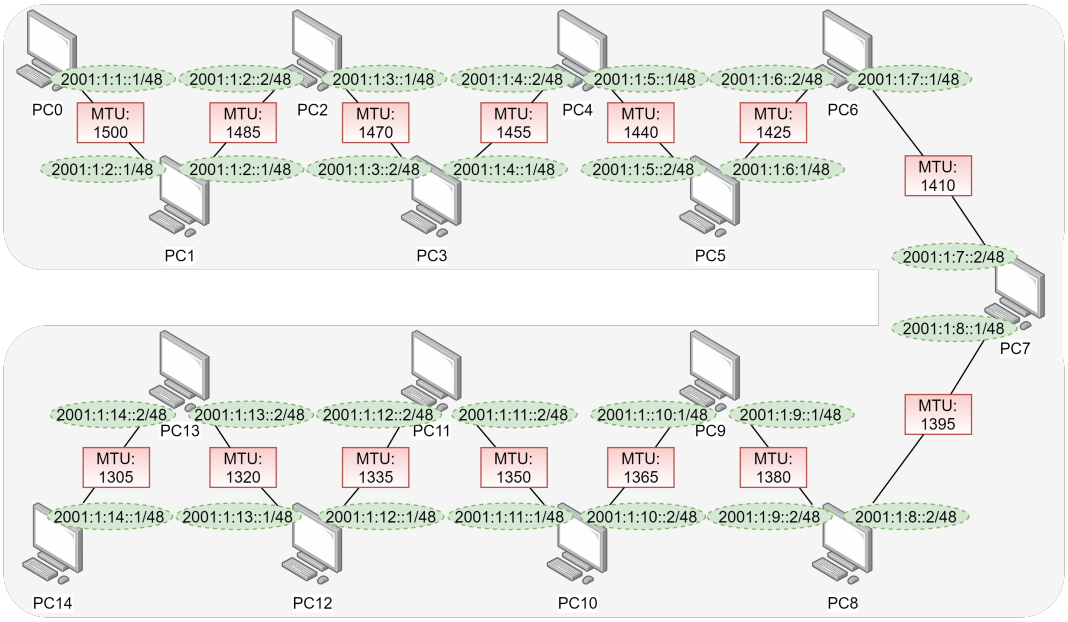

Figure 2. Proposed IPV6 Scenario. Source: Elaborated by the Authors.

Every network device is connected to an internal network, where PC0 was connected with adapter 1 to Internal Network sw1, with IP 2001:1:1::1/48 configuration, default MTU 1500, and default gateway 2001:1:1::2/48.

PC1, in service, was connected with adapter one on the Internal Network sw1 and adapter two on the Internal Network sw2, IP configuration: 2001:1:1::2/48 on eth0 and standard MTU of 1500, and IP 2001:1:2::2/48 on eth1 with MTU of 1485, and with default gateway 2001:1:2::1/48. Enabling packet routing on the machine, uncommenting the line: net.ipv4.ip_forward = 1 and the line net.ipv6.conf.all.forwarding = 1 in the /etc/sysctl.conf file. Additionally, adding the following commands in /etc/rc.local, to change the MTU value (ifconfig eth1 MTU 1485), and command to activate NAT (ip6tables -t nat -A POSTROUTING -o eth0 -j MASQUERADE / ip6tables -t nat -A POSTROUTING -o eth1 -j MASQUERADE) to allow you to rewrite the source IP addresses of a packet that passed over a router so that a computer from an internal network had access to an external network.

The configurations of the other machines will follow that of the PC1 described above alternating the Internal Networks of the adapters, the values of IP, MTU, and the addition of the IP of the next machine as a gateway, by their respective, of Figure 2.

In PC14, the configuration was similar to that of PC0. Where PC14 was connected with adapter one on the Internal sw14 network. IP configuration 200:1:14::1/48, MTU 1305, and with default gateway 2001:1:14::2/48. Add the command in /etc/rc.locla, to change the MTU value (ifconfig eth0 MTU 1305).

Once these settings were made, PC0 had its connection established with PC14 through each machine, as contained the topology of Figure 2.

4. Results

With the setup of the scenario in both the IPv4 network protocol and the IPv6 network protocol, the stage of testing packet forwarding and collecting performance results in the network flow began, and comparisons were made between the obtained results and presented as conclusions regarding the efficiency of the IPv6 Path MTU Discovery protocol.

4.1. Tests in the IPv4 Scenario

Therefore, with the scenario in the IPv4 network protocol ready for the packet forwarding tests for network flow analysis, which were carried out with Iperf, a tool suggested by RFC 6349 (2011). The study of network performance allowed for measuring the bandwidth, evaluating the latency, the jitter (variation of the delay), and the quality of the link between the points, according to Filho [18].

The tool operated in client-server mode, with a machine that worked as a server at one end of the network, which received requests and, at the other end, a device that worked as a client, which generated all network traffic. Therefore, PC0 with the highest MTU (1500) assumed the role of client and PC14 with the lowest MTU (1305) considered the part of the server, the parameters used in PC14 were iperf -u -s and the -u to use the UDP protocol and -s to assume the server mode.

In PC0, the parameters used were iperf -c 10.0.14.2 -u -i 1 -n 100K -b 10G, with -c to assume the client mode. 10.0.14.2 being the IP of the server that received the requests, -u to use the UDP protocol, -i 1 to determine the time of 1 s to display the output report, -n 100K to determine the number of bytes for transmission and -b 10G to determine the 10G band, as shown in Figure 3.

```
root@debian:~# iperf -c 10.0.14.2 -u -i 1 -n 100K
Client connecting to 10.0.14.2, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KBytes (default)
[3] local 10.0.1.1 port 52081 connected with 10.0.14.2 port 5001
[ID] Interval     Transfer    Brandwidth
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec
[3] Send 70 datagrams
[3] Server Report:
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec 0.173ms     0/ 70  (0%)
```

Figure 3. Screenshot of "iperf" command running on the IPv4 Client.

In PC14, the values of transfer sent, where the value of 100K is required in the client device, the amount of bandwidth, and jitter, as shown in Figure 4.

```
root@debian:~# iperf -s -u
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KBytes (default)
[3] local 10.0.1.1 port 5001 connected with 10.0.14.1 port 52081
[ID] Interval     Transfer    Brandwidth     Jitter  Lost/Total Datagrams
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec 0.174ms     0/ 70  (0%)
```

Figure 4. Screenshot of "iperf" results on the IPv4 Server.

4.2. Tests in the IPv6 Scenario

With the scenario in the IPv6 network protocol, the tests were performed following the same model that was used in IPv4, only changing the iperf parameters, which, in PC14, were iperf -u -s -V, with -u to use the UDP protocol, -s to assume server mode, and -V to use in IPv6 mode. In PC0, the parameters used were iperf -u -i 1 -V -c 2001:1:14::1 -n 100K -b 10G, -u to use the UDP protocol, -i 1

to determine the time of 1 s to display the output report, -V for use in IPv6 mode, 2001:1:14::1 being the IP of the server that received the requests, -n 100K to determine the number of bytes for transmission, and -b 10G for determining the 10G band, as shown in Figure 5.

```
root@debian:~# iperf -u -i 1 -V -c 2001:1:14::1 -n 100K
Client connecting to 2001:1:14::1, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 160 KBytes (default)
[3] local 2001:1:14::1 port 56405 connected with 2001:1:14::1 port 5001
[ID] Interval       Transfer     Brandwidth
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec
[3] Send 70 datagrams
[3] Server Report:
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec 0.220ms     0/ 70  (0%)
```

Figure 5. Screenshot of "iperf" command running on the IPv6 Client.

In PC14, the values of transfer, which the value at 100K is required in the client device, returned the value of bandwidth and jitter, as shown in Figure 6.

```
root@debian:~# iperf -s -u -V
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 160 KBytes (default)
[3] local 2001:1:14::1 port 5001 connected with 2001:1:14::2 port 56405
[ID] Interval       Transfer     Brandwidth       Jitter  Lost/Total Datagrams
[3] 0.0-0.8 sec   100KBytes   1.05 Mbits/sec 0.220ms     0/ 70  (0%)
```

Figure 6. Screenshot of "iperf" results on the IPv6 Server.

4.3. Analysis of Results

The tests were carried out with iperf, changing the number of bytes for transmission, which started at 100KBytes, then switched to 200KBytes, 300KBytes, and so on with jumps from 100 to 100KBytes, until it reached the value of 2000KBytes, and then tests with 5MBytes, 10MBytes, and 50MBytes.

For a better analysis of the results, five tests were performed with each value of transmission bytes. An average of the five results was calculated, so that all variations that could occur were considered, and bring the closest to reality and stable values.

With the obtained results, it was possible to assemble the following graphs for a better view of each protocol's performance, with the scenario of fragmented MTU and standard MTU. In Figure 7, it is possible to observe the relationship of the bandwidth and jitter of the tests performed in the IPv4 scenario with the fragmented MTU. Low bandwidth and relatively high jitter are noted.

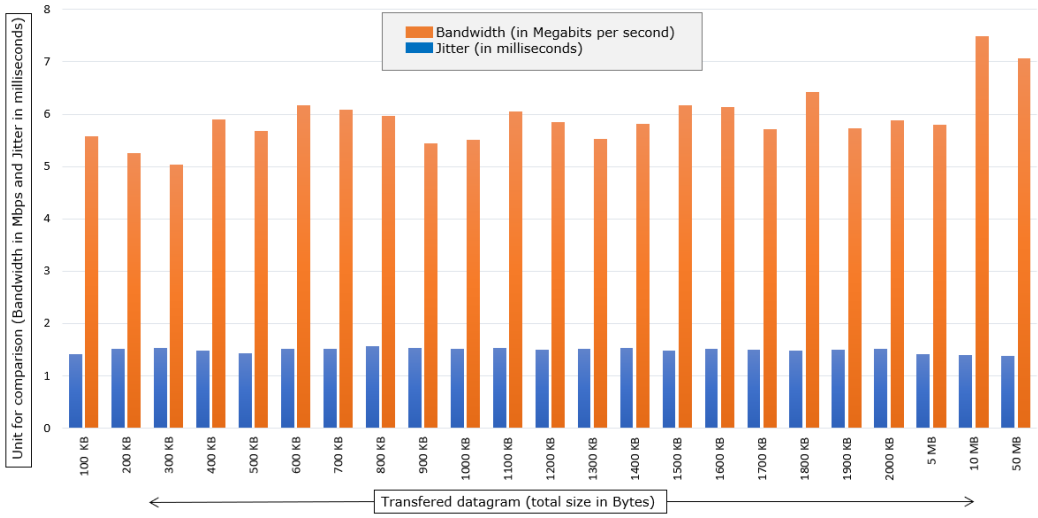

Figure 7. Results in the IPv4 Scenario with Fragmentation. Source: Elaborated by the Authors.

In Figure 8, it is possible to observe the relationship between the bandwidth and jitter of the tests that were performed in the IPv4 scenario with standard MTU. You can see a higher bandwidth and a relatively lower jitter than the previous scene.

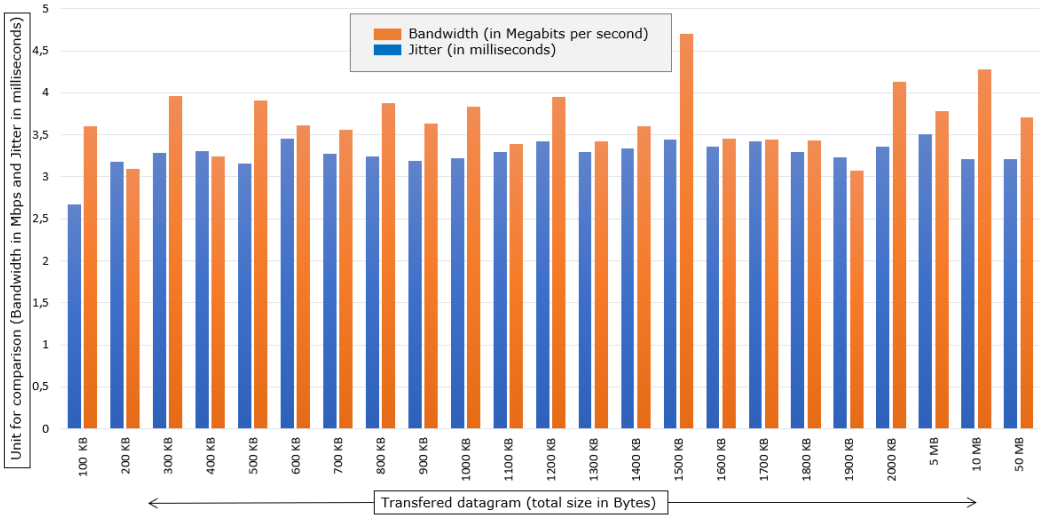

Figure 8. The results in the IPv4 Scenario without Fragmentation. Source: Elaborated by the Authors.

In Figure 9, it is possible to observe the relationship of the bandwidth and jitter of the tests performed in the IPv6 scenario with the fragmented MTU. A smaller bandwidth and a relatively higher jitter are noted than the previous scene.

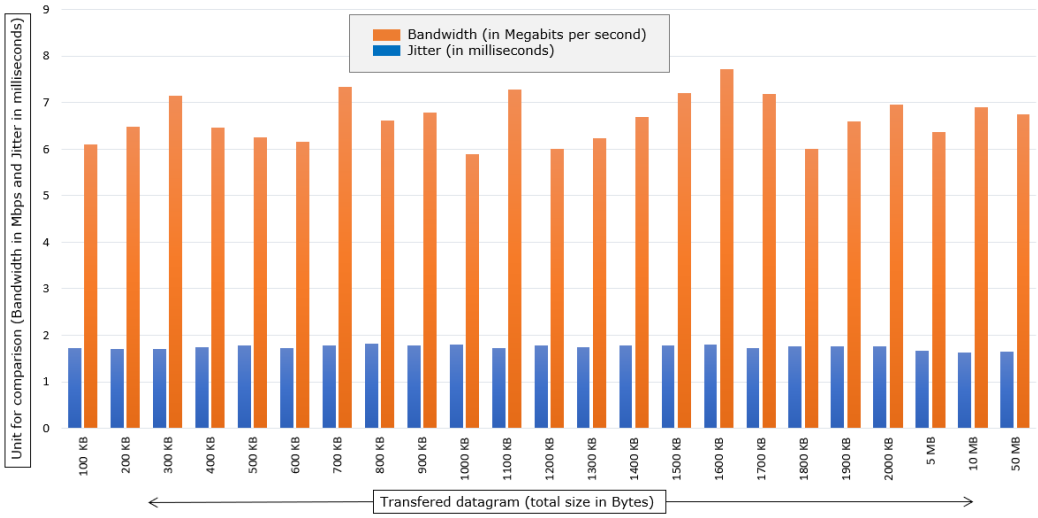

Figure 9. Results in the IPv6 Scenario with Fragmentation. Source: Elaborated by the Authors.

In Figure 10, it is possible to observe the relationship of the bandwidth and jitter of the tests performed in the IPv6 scenario with the standard MTU. Note a bandwidth and jitter similar to the previous scene.

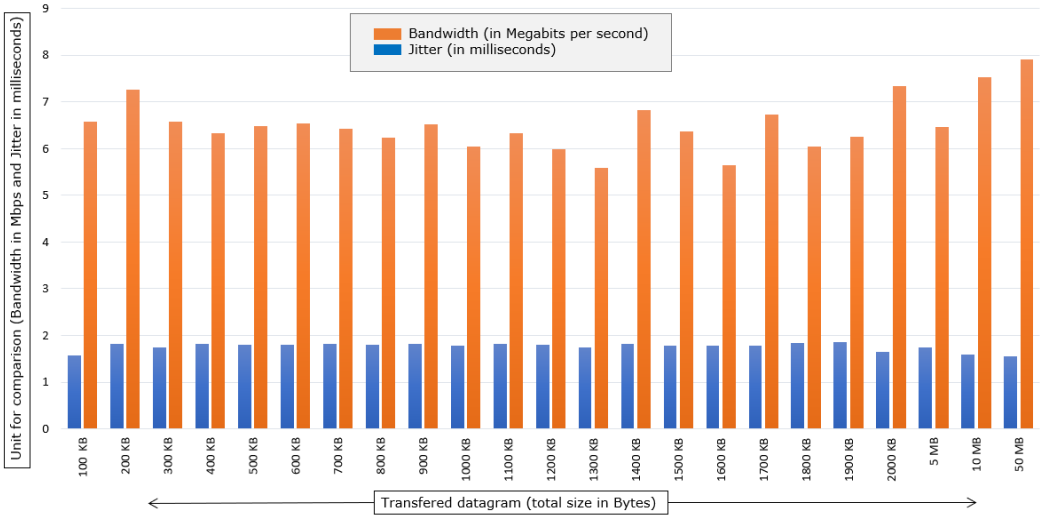

Figure 10. Results in the IPv6 Scenario without Fragmentation. Source: Elaborated by the Authors.

In Figure 11, it is possible to observe the relationship between the IPv4 and IPv6 scenarios with the standard and fragmented MTU, divided by the red frames. The values point to a stabilization in bandwidth and jitter when Path MTU Discovery is used.

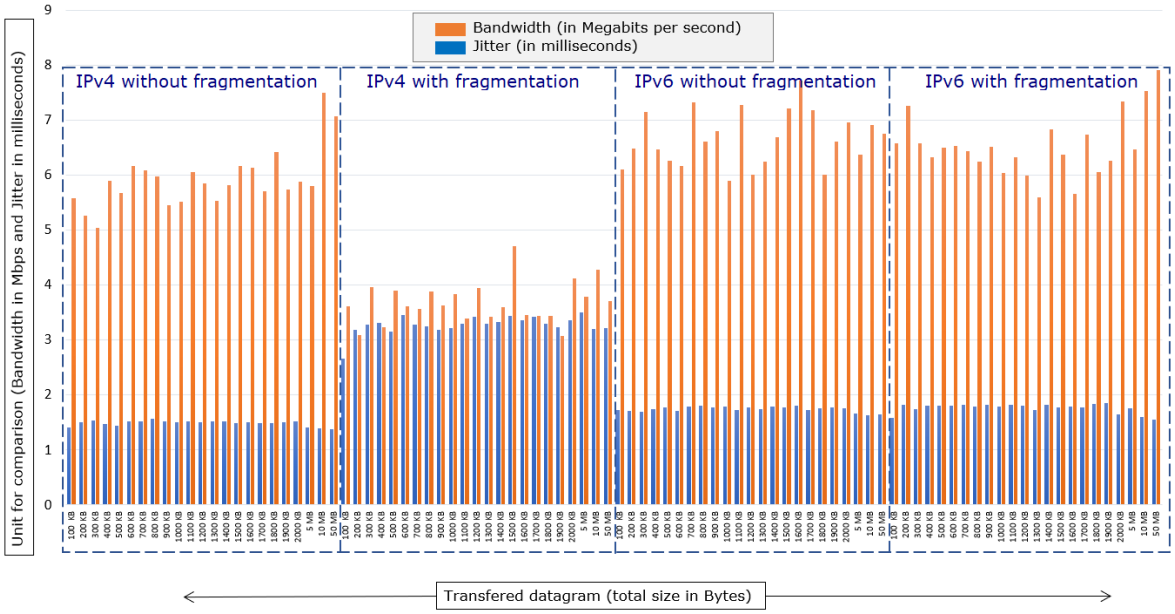

Figure 11. General Comparison of Results. Source: Elaborated by the Authors.

5. Conclusions

The elaboration of this work allowed for an analysis of the fragmentation of IP packets in IPv4 and IPv6 network environments, and to present the functionality of the Path MTU Discovery protocol that is integrated to IPv6.

Some objectives were defined for better analysis in obtaining the expected results; the literature reviews the association of packet fragmentation in IPv4 networks whose internet can present different MTU sizes. The literature review of the operation of the Path MTU Discovery protocol in IPv6 networks creates a virtual environment that presents different MTU sizes on its nodes and performs performance tests while using UDP transport traffic through IPv4 networks and IPv6 networks.

The work also used searches in articles, books, and websites in the area of computer networks for bibliographic review, and the VirtualBox software for the creation of fragmented scenarios, which made it possible to carry out tests with Iperf, and allowed for the network performance analysis, measuring bandwidth, latency, and jitter (delay variation).

Through packet forwarding tests and collection of performance results in the network flow, it can be concluded that the values point to a stabilization in bandwidth and jitter when Path MTU Discovery is used, in both a fragmented IPv6 scenario as in a typical IPv6 scenario. Still, in an IPv4 situation that does not have Path MTU Discovery, there is the change between fragmented and standard scenes, proving that the use of the Path MTU Discovery protocol, which is inherent to IPv6, is efficient in relation to traffic performance in similar tests on IPv4 networks that presented fragmentation scenarios.

Author Contributions: Formal analysis, T.L., M.F., R.P., G.F. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Comer, D. *Networking with TCP / IP–: Principles, Protocols and Architecture*; Elsevier Editora Ltd.: Amsterdam, The Netherlands, 2016; Volume 1.
2. Torres, G. *Computer Network*; Novaterra: Pamplemousses, Mauritius, 2009.
3. Google. Statistic, 2019. Available online: https://www.google.com/intl/pt-BR/ipv6/statistics.html (accessed on 10 February 2020).

4. Luckie, M.; Stasiewicz, B. Measuring Path MTU Discovery Behaviour. In Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, Melbourne, Australia, 1–3 November 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 102–108. [CrossRef]
5. Narayan, S.; Kolahi, S.S.; Sunarto, Y.; Nguyen, D.D.T.; Mani, P. Performance comparison of IPv4 and IPv6 on various windows operating systems. In Proceedings of the 2008 11th International Conference on Computer and Information Technology, Khulna, Bangladesh, 24–27 December 2008; pp. 663–668. [CrossRef]
6. Narayan, S.; Shang, P.; Fan, N. Network performance evaluation of Internet Protocols IPv4 and IPv6 on operating systems. In Proceedings of the 2009 IFIP International Conference on Wireless and Optical Communications Networks, Cairo, Egypt, 28–30 April 2009; pp. 1–5. [CrossRef]
7. Li, Q.; Qin, T.; Guan, X.; Zheng, Q. Empirical analysis and comparison of IPv4-IPv6 traffic: A case study on the campus network. In Proceedings of the 2012 18th IEEE International Conference on Networks (ICON), Singapore, 12–14 December 2012; pp. 395–399. [CrossRef]
8. Ali, A. Comparison study between IPV4 & IPV6. In *IJCSI International Journal of Computer Science Issues*; Philadelphia University: Philadelphia, PA, USA, 2012; Volume 9.
9. Cabellos-Aparicio, A.; Julian-Bertomeu, H.; Núñez-Martínez, J.; Jakab, L.; Serral-Gracià, R.; Domingo-Pascual, J. *Measurement-Based Comparison of IPv4/IPv6 Mobility Protocols on a WLAN Scenario*; UPC: Barcelona, Spain, 2005.
10. Yasinovskyy, R.; Wijesinha, A.L.; Karne, R.K.; Khaksari, G. A comparison of VoIP performance on IPv6 and IPv4 networks. In Proceedings of the 2009 IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Morocco, 10–13 May 2009; pp. 603–609. [CrossRef]
11. Hyun, J.; Li, J.; Kim, H.; Yoo, J.; Hong, J.W. IPv4 and IPv6 performance comparison in IPv6 LTE network. In Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, Korea, 19–21 August 2015; pp. 145–150. [CrossRef]
12. Luckie, M.; Cho, K.; Owens, B. Inferring and Debugging Path MTU Discovery Failures. In Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement, Berkeley, CA, USA, 19–21 October 2005; p. 17.
13. Göhring, M.; Shulman, H.; Waidner, M. Path MTU Discovery Considered Harmful. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 866–874. [CrossRef]
14. Xiao, R.; Wei, Y.; Xiao, Y.; Zhu, X.; Sun, B. PFPG: A novel MTU model for IPv4/IPv6 transition. In Proceedings of the 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Xiamen, China, 19–21 August 2014; pp. 854–859. [CrossRef]
15. Custura, A.; Fairhurst, G.; Learmonth, I. Exploring Usable Path MTU in the Internet. In Proceedings of the 2018 Network Traffic Measurement and Analysis Conference (TMA), Vienna, Austria, 26–29 June 2018; pp. 1–8. [CrossRef]
16. Supriyanto, S.; Sofhan, R.; Fahrizal, R.; Osman, A. Performance evaluation of IPv6 jumbogram packets transmission using jumbo frames. In Proceedings of the 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, 19–21 September 2017; pp. 1–5. [CrossRef]
17. Praptodiyono, S.; Sofhan, R.; Pramudyo, A.S.; Firmansyah, T.; Osman, A. Performance comparison of transmitting jumbo frame on Windows and Linux system. *Telkomnika (Telecommun. Comput. Electron. Control.)* **2019**, *17*, 68–75. [CrossRef]
18. Filho, J. *Traffic Analysis on TCP/IP Networks*; Novatec: Paris, France, 2013.