Performance Comparison of IPv6 in 802.11ac WLAN in Windows and Linux Environment

Apaula Aotearoa Tuifaiga, Priyanka Ashreena Ram, Samad Salehi Kolahi,
Unitec Institute of Technology, New Zealand
aotearoagale@gmail.com
priyankreena8@gmail.com
skolahi@unitec.ac.nz

Abstract **- In this paper, the performance of IEEE 802.11ac wireless LAN is analysed and compared for two operating systems, Windows, and Linux Ubuntu. Parameters measured are throughput, latency, and CPU utilization for TCP, UDP, IPv4, and IPv6. For both wireless 802.11ac network implementations studied, client-server and peer-to-peer, our results shows that the throughput for both Windows and Ubuntu is much higher in IPv4 than IPv6. IPv6 also has higher RTT and higher CPU utilization than IPv4. TCP has lower bandwidth than UDP, and Windows had lower throughput than Linux. The highest throughout achieved in systems considered is for client server with IPv4 and UDP at throughput of 757Mbps for Windows and 798Mbps for Ubuntu. For channel size of 80MHz we could not reach theoretical limit of 1.3Gbps for wireless LAN 802.11ac.**

Keywords: *IEEE 802.11ac, IPv6, Windows, Linux, Peer-to-Peer, Client Server, Performance Analysis*

I. Introduction

Wireless LAN is very popular these days. The use of wireless connection has made it much easier for not only businesses, but the wider community to connect with others without the hassle of using wired or physical connection to access the network/internet. Using wireless technology has become a part of our daily life, therefore, it is very important and a must, to evaluate and analyse the performance of a wireless network using different network set ups.

IEEE 802.11ac is the newest standard of the IEEE 802.11 suite/family. Evaluating this new standard is very important as many businesses are using IEEE 802.11ac. IEEE 802.11ac is an improvement of IEEE 802.11n as it has more advanced features like beamforming, increased number of spatial MIMO streams and extension of channel widths. IEEE 802.11ac's focus is to give its users the best performance. Theoretically, IEEE 802.11ac should reach up to 1.3Gbps while IEEE 802.11n provides up to 600Mbps [1]. In this study we will also evaluate this theoretical limit.

The connectivity and performance of devices on a network also depends on the architecture and operating system platform used. The most common network architectures that are being used up to date are; client-server and peer-to-peer and we study both. We will use two operating systems, Linux and Windows. The performance of latest Internet protocol IPv6 is also compared with IPv4 protocol.

II. Literature Review

In 2010 [2], authors compared the performance of IPv4 and IPv6 in a peer-to-peer and client-server local area network for a wired network. The result of their experiment showed that IPv4 produced higher bandwidth for TCP protocol. The results also showed than client server is faster than peer-peer for a wired LAN.

In 2011 [3], authors analyzed the influence WPA2 security has on IPv4 and IPv6 using 802.11n wireless LAN. The results showed that the maximum bandwidth difference between an open system (no security) and a system with WPA2 implemented. WPA2 security slowed the network.

In 2011 [4], a research was carried out by authors who evaluated the performance on the latest version of the Internet protocol (IPv6) in a peer-to-peer environment 802.11n WLAN. This was done on Windows XP, Windows 7 and Fedora 12 using 802.11n which were the latest developments at the time. Authors also analyzed the performance of IPv6 in comparison to IPv4 on a wireless peer-to-peer 802.11n with WPA2 environment They measured RTT, throughput, and CPU utilization.

In 2012 [5], authors evaluated TCP throughput using various packet sizes to test the performance of IEEE 802.11n in a client-server environment. Both open system and system with WPA2 implemented was considered and compared. Their results showed that Windows Vista IPv4 reached the highest TCP throughput of 120Mbps when with no security implemented and 110Mbps when security included.

In 2015 [6], authors investigated the bandwidth of IEEE 802.11ac in comparison to IEEE 802.11n in a client-server environment. Their testbed consisted of a single client, access point, and a server. Client and access point were set apart 10m from each other. Iperf was then used to measure the performance in terms of TCP bandwidth.

In 2015 [7], a research was conducted by authors to evaluate the performance of IEEE 802.11ac in comparison to IEEE 802.11n using both versions of the Internet Protocol (IPv4 and IPv6) on two Windows 7 machines. This experiment was done in a wireless peer-to-peer environment. The results of this research concluded that 802.11ac throughput for both IPv4 and IPv6 is much higher than 802.11n. In terms of jitter, IPv4 was higher than IPv6 in both 11ac and 11n.

In 2017 [8], authors evaluated the impact of WPA2 security on wireless client-server bandwidth and latency network using the IEEE802.11ac standard.

In 2019 [9], authors investigated the Impact of Human Shadowing/Movement on Performance of 802.11ac Client-to-Server WLAN.

There has been no research to this date in terms of comparing IPv6 with IPv4 for both client-server and peer-to-peer, Linux and Windows, for both IPv4 and IPv6, and TCP and UDP. The motivation behind this work is therefore to obtain new results in a real world test bed environment for 802.11ac Wireless LAN for both peer-peer and client server, comparing IPv4 and IPv6, TCP and UDP.

III. Network Setup

We first setup a peer-to-peer network that contains two PCs both with Intel® Core™ i7-6700 processors at 3.40GHz with RAM of 16.0GB. These PCs were wirelessly connected to a wireless access point, Linksys LAPAC1750 Pro that was configured to run at 80MHz. These two PCs were wirelessly connected to the access point (LAPAC1750 Pro).

We then set up a client-server network. This setup consists of one PC with Intel® Core™ i7-6700 processors at 3.40GHz with RAM of 16.0GB with Windows 10 installed. This PC acted as the client. A Windows Server was also installed in another PC. The server was connected to the access point (LAPAC1750 Pro) using a straight through cable. This access point is also configured with 80MHz channel width. Our testbeds were setup within few meters distance from each other so the maximum throughput, round trip time and CPU utilization could be achieved. We then repeated the above process for the Ubuntu Linux testing, setting up peer-peer and client server. The tests continued for both IPv4 and IPv6 with TCP or UDP at the transport layer.

IV. Data Generation and Measurement Tools

D-IGT [10] and Netperf [11] were both selected as the generating and measurement tools for our experiment. We chose to use two different tools to check and verify the consistency of our data. D-ITG was used to measure throughput as well as Netperf. With D-ITG we managed to determine the round trip time. Netperf was used to determine the CPU utilization. When generating data, we used a 1-minute interval for each run, and did 10 runs per packet size, and then averaged the results of runs that gave us results with 95% confidence interval. D-ITG and Netperf are commonly used tools for performance evaluation as seen on pervious work cited in references.

For the performance measurement, throughput, round trip time and CPU usage was tested for TCP, UDP, IPv4 and IPv6, in both Peer-to-Peer and Client-Server network setups. Different packets sizes were used ranging from 128, 256, 384, 512, 640, 768, 896, 1024, 1152 to 1408 Bytes. This measurement procedure was carried out on all tests we performed.

V. Results

In this section, first the results are presented, then we discuss the reason behind some of the results.

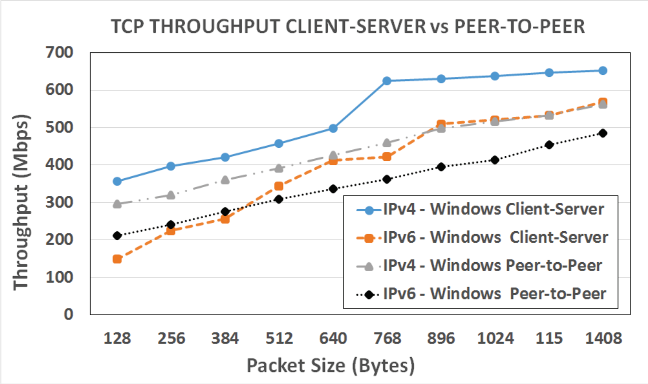

Figure 1: Windows TCP Throughput

Figure 1 shows the result of TCP throughput in both the client-server and Peer-to-peer environment. In the client-server environment, IPv4 reached the highest TCP throughput of 652Mbps for all packet sizes tested, while IPv6 reached 568Mbps. In the peer-to-peer environment, IPv4 also gave the highest TCP throughput of 562Mbps and IPv6 with 486Mbps which happens to be the lowest. Of both network environment, IPv4 produced the highest TCP throughput and IPv6 with the lowest due to high overhead in TCP. For client-server, IPv4 is much higher than IPv6 by 84Mbps difference. For peer-to-peer, IPv4 is also higher by 76Mbps than IPv6.

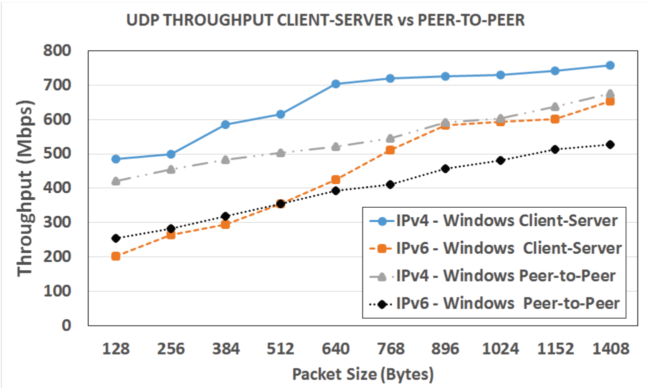

Figure 2: Windows UDP Throughput

Figure 2 shows the UDP throughput for client-server and peer-to-peer. IPv4 client-server clearly has the highest throughput of 757Mbps, while IPv6 peer-to-peer has the lowest of 528Mbps. For both network environment, IPv4 has the highest throughputs. There is an increase in throughput of IPv6 in client-server environment as the packet sizes increase. At packet size 512 Bytes, IPv6 throughput for both client server and peer to peer reached almost the same throughput with IPv6 peer to peer being the highest by 1Mbps (client server; 354Mbps, peer-to-peer; 355Mbps). For client-server, IPv4 is much higher than IPv6 by 103Mbps. For peer-to-peer, IPv4 is also higher by 147Mbps than IPv6. This clearly shows that IPv4 for both testbed environment has much better performance than IPv6. Comparing TCP and UDP for client-server, IPv4 UDP gave the highest throughput of 757Mbps and IPv6 TCP with the lowest throughput of 568Mbps. For both IPv4 and IPv6, UDP has the highest throughput. For peer-to-peer, IPv4 UDP also gave the highest throughput of 675Mbps with IPv6 TCP the lowest at 486Mbps. Overall

y.

UDP has the highest throughput performance than UDP for both client-server and peer-to-peer. This is due to lower overheads in UDP. IPv4 had higher bandwidth than IV6 due to high overheads in IPv4 packet (128 bit for addressing).

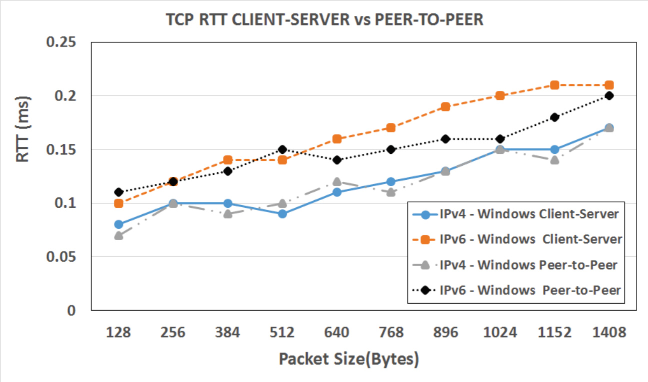

Figure 3: Windows TCP Round Trip Time

Round trip time for TCP traffic was recorded and is shown in Figure 3. Client-Server IPv6 shows the highest RTT recorded followed by IPv6 in peer-to-peer. Looking at the results there is not that much difference between IPv4 RTT in both client-server and peer-to-peer. For IPv6 it is clear that client-server has higher RTT than peer-to-peer except for packet size 512 where peer-to-peer is negligibly higher than client-server by 0.01ms. This can be due to margin of errors, or different interference from other sources at measurement times at the access point.

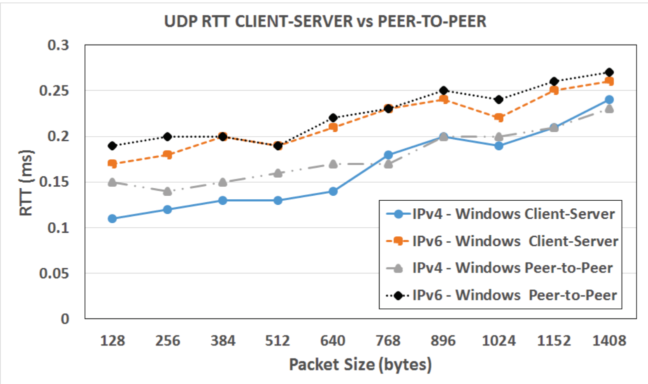

Figure 4: Windows UDP Round Trip Time

For UDP, IPv6 gave the highest round trip time for both client-server and peer-to-peer networks (Figure 4). Again, the round trip time increases as the packet size increased. IPv6 RTT for client-server and peer-to-peer seem to be close with each other. At packet sizes 384 and 512 Bytes, both client-server and peer-to-peer IPv6 gave the RTT of 0.2ms and then dropped down to 0.19ms. At packet size 768 Bytes, they both provided 0.23ms which are quite higher than IPv4. For IPv4, client-server had the lowest round trip time till it reached the packet size of 768 Bytes where it gave a round trip time of 0.18ms which is an increase from the round trip time of 0.12ms at packet size 640Bytes. At packet size 896 Bytes, IPv4 provide the same round trip time for both client-server and peer-to-peer and at packet size 1024 Bytes, the client-server's TCP round trip time dropped down to 0.19ms and then increased in the last two packets sizes of 1152 and 1408 Bytes. But these could be because of margin of errors or different interference to access point at measurement time.

IPv4 gave the lowest round trip time in a peer-to-peer environment compared to IPv6 (due to largest packet size of IPv4). Looking at both RTT for TCP and UDP, for client-server itself, IPv6 UDP gave the highest round trip time of 0.26ms while IPv4 TCP gave the lowest of 0.17ms. For peer-to-peer, IPv6 UDP also gives the highest round trip time of 0.27ms and IPv4 TCP with the lowest of 0.17ms. Overall, IPv6 UDP has the highest round trip time for all packet sizes and for both client-server and peer-to-peer networks, due to high packet overhead size in IPv6.

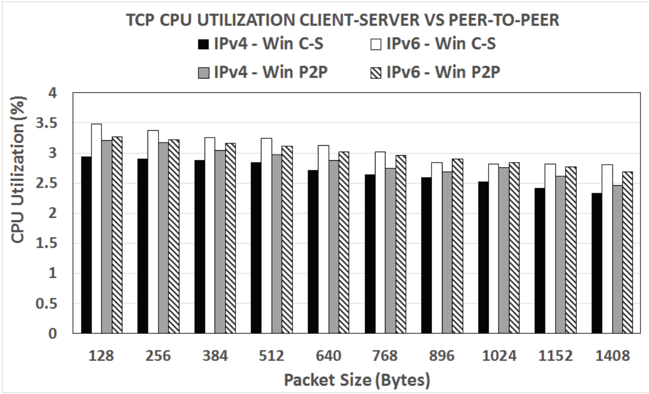

Figure 5: Windows TCP CPU Utilization

TCP CPU utilization for both client-server and peer-to-peer are as shown in Figure 5. IPv6 shows the higher usage than IPv4 for both client server and peer to peer. Due to high overhead, IPv6 in client-server has the highest CPU usage for all packets except for packet size 896 where IPv6 in peer-to-peer usage increased to 2.90% and became slightly higher than client-server (2.84%). For IPv4, peer-to-peer has the higher CPU usage than client-server; the highest being at 3.21% and client-server at 2.94%. Overall, IPv6 uses the most CPU for both client-server and peer-to-peer due to higher overhead.

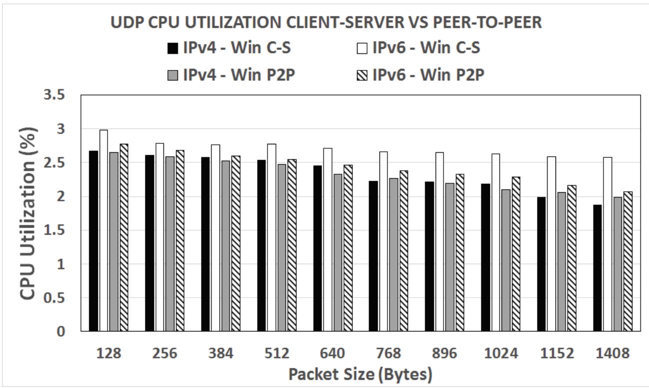

Figure 6: Windows UDP CPU Utilization

For UDP CPU Utilization (Figure 6), IPv6 is much higher than IPv4 for all packet sizes and for both client-server and peer-to-peer. The highest CPU usage was recorded at packet 128 Bytes for IPv6 and client-server. The lowest CPU usage of 1.87% was recorded at the largest packet size of 1408Bytes for IPv4 client-server. For IPv6 itself, client-server gave the highest CPU usage of 2.98% and peer-to-peer at 2.77%. For IPv4, client-serve also gave the higher CPU usage expect for packet sizes 768 and 1152 Bytes where peer-to-peer gave the higher CPU usage for both client server and peer-to-peer.

Comparing TCP and UDP CPU usage for client-server, IPv6 TCP has the highest CPU usage of 2.81% and IPv4 UDP with the lowest of 1.87%. For peer-to-peer, IPv6 TCP has the highest of 2.69% CPU usage and 1.99% usage for IPv4 UDP which is also the lowest CPU utilization usage recorded for peer-to-peer. Overall, IPv6 TCP for both client-server and peer-to-peer gave the highest CPU usage.

The CPU usage was higher at low packet sizes as there are many packets to be processes with small packets.

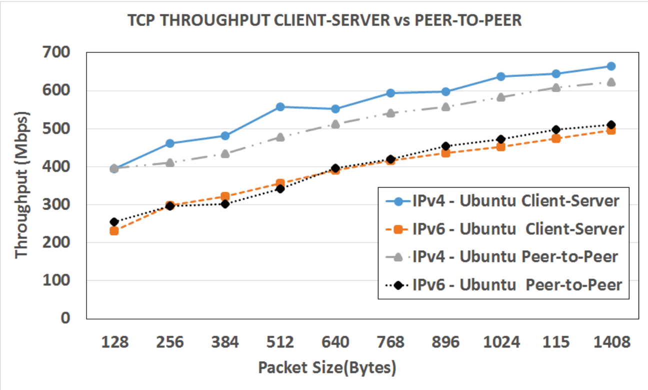

Figure 7: Ubuntu TCP Throughput

Figure 7 shows the results of Ubuntu's TCP throughput for both IPv4 and IPv6. IPv4 reached a maximum throughput of 664Mbps and IPv6 at 496Mbps. This gives a difference of 168Mbps between the two protocols at the largest packet size of 1408Bytes. The highest difference between the two was point for client-server was 200Mbps at packet size 512 Bytes. For IPv4, the throughput increases as the packet size increases except for packet size 640bytes. At packet 512 Bytes, throughput reached 557Mbps then drops by 5Mbps at packet 640 Bytes (552Mbps).

For peer-to-peer, IPv4 gives the highest throughput of 622Mbps and IPv6 has 511Mbps, giving a difference of 111Mbps at the highest packet size of 1408 Bytes. Just by looking at the figure above there is a massive difference between IPv4 and IPv6, not only at the highest packet size, but with all other packet sizes. The highest difference point between IPv4 and IPv6 is for the peer-to-peer setup was 143Mbps at packet size 128 (IPv4 was 397Mbps, IPv6 – was 254Mbps). Comparing the two setups, IPv4 client-server has the higher maximum throughput (664Mbps) compared topeer-peer. IPv6 client-server and IPv6 peer-to-peer are not too far away from each other. At packet sizes 256, 384 and 512 Bytes, IPv6 client-server was higher than IPv6 peer-to-peer. From 128, 640, 768, 896, 1024, 1152 and 1408 Bytes IPv6 peer-to-peer was higher.

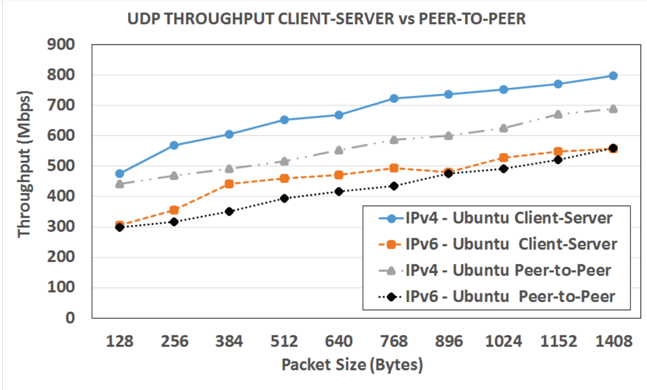

Figure 8: Ubuntu UDP Throughput

The above figure (Figure 8) shows the results of UDP throughput. For client-server, IPv4 has the highest throughput of all packet sizes and IPv6 with the lowest on all packet sizes. IPv4 reached the highest throughput of 798Mbps while IPv6 was 557Mbps. The highest difference point between IPv4 and IPv6 was 255Mbps at packet size 896 Bytes.
As the figure illustrates, in the peer-to-peer environment, IPv4 also has the higher throughput of 690Mbps. IPv6 reached the maximum throughput of 560Mbps giving the difference of 130Mbps between IPv4 and IPv6. The highest difference point for a packet size was 152Mbps at packet size of 256 Bytes.

Overall, IPv4 client-server shows the highest throughput (798Mbps) and IPv6 peer-to-peer being the lowest (557Mbps). Client server is faster than peer-peer as it has a server that makes the network faster.

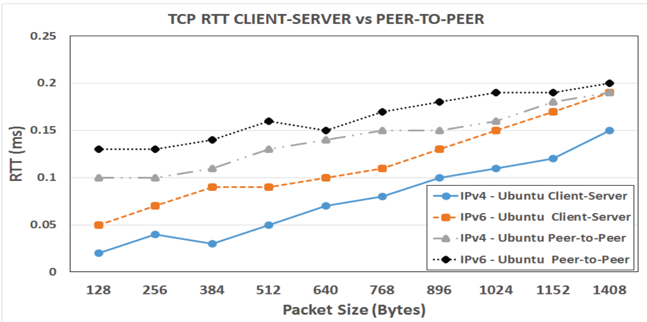

Figure 9: Ubuntu TCP Round Trip Time

Figure 9 shows the TCP round trip time for both IPv4 and IPv6. For the client-server setup, IPv6 has the highest RTT of 0.19ms at the largest packet size and IPv4 at 0.15ms. As the packet size increases, RTT also increases. For IPv4 client-server, round trip time increases at the packet size increase except for packet size 384 where the round trip time is 0.03ms which is a 0.01ms drop from the previous packet. For IPv6 the round trip constantly increases as the packet size increases. At packet size 384 and 512, the round trip time remains the same at 0.09ms.

For peer-to-peer, IPv6 also has the highest round trip time and IPv4 the lowest. The maximum difference between the two is 0.01ms at the largest packet size (IPv6 - 0.2, IPv4 – 0.19).

Overall, IPv4 client-server has less round trip time of all packet sizes and for both setups (0.15ms) outperforming IPv6 peer-to-peer which has the highest round trip time of 0.2ms, giving a difference of 0.05ms between the two.

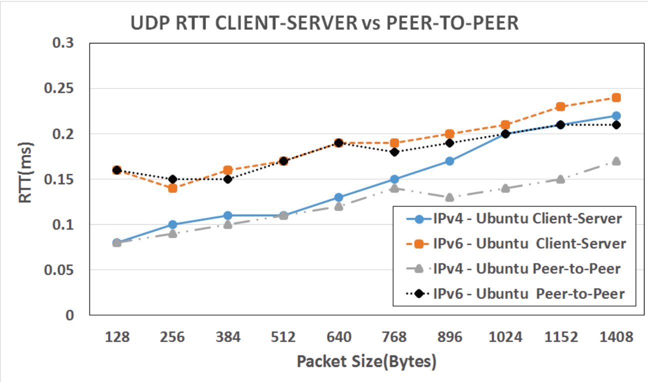

Figure 10: Ubuntu UDP Round Trip Time

The above figure illustrates the results of UDP round trip time for IPv4 and IPv6. For client-server, IPv4 has less round trip time than IPv6. The highest time recorded was 0.24ms for IPv6 and 0.22ms for IPv4. For IPv4, round trip time increases as the packet size increase and at some points, it remains the same (packet 384 and 512). For IPv6 the round trip time also increases except for packet 256 where its less by 0.02ms than both 128 and 384 (0.16ms).

For peer-to-peer, IPv4 has less round trip time than IPv6 (0.17 and 0.21). IPv4 clearly outperformed IPv6 as the maximum difference between the two was 0.04ms at the largest packet size. For IPv4, there are points where round trip time remains the same (packet sizes 768 and 1024 Bytes) where they were both at 0.14ms. For packet size 896 Bytes where round trip time is less by 0.01ms. For IPv6, packets 256 and 384 Bytes both remain at 0.15ms as well as packets 1152 and 1408 at 0.21ms.

For both client-server and peer-to-peer networks, IPv4 client-server outperformed IPv6 with peer-to-peer network meaning IPv4 is faster as it has lower overhead, and client–server is faster as it had a server.

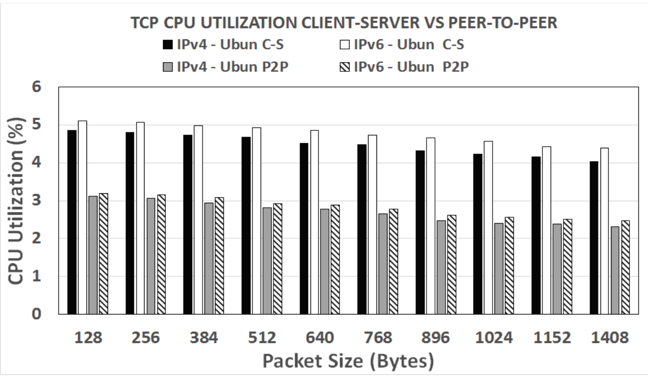

Figure 11: Ubuntu TCP CPU Utilization

Figure 11 showed the TCP CPU Utilization for both IPv4 and IPv6 for client-server and peer-to-peer networks. For client-server, the highest CPU usage was 5.1% for IPv6 at packet size 128 Bytes. IPv4 had the least CPU usage. For the largest packet size (1408), IPv6 has the highest CPU of 4.38% with a difference of 0.35% from IPv4.

For peer-to-peer, IPv6 had the highest CPU utilization of 3.18%. IPv4 had the lowest of 3.11% at packet size 128bytes. At packet size 1408Bytes, IPv6's CPU usage was 2.47% with IPv4 being less by 0.15%.

IPv6 clearly had the most CPU utilization with IPv6 client-server at 5.1% and IPv4 peer-to-peer at 3.18% at packet size 128Bytes. With all other packet sizes CPU utilization decreases consistently as the packet size increases for both networks. Bigger packet sizes mean less packets to process and overall less overhead, resulting in lower CPU.

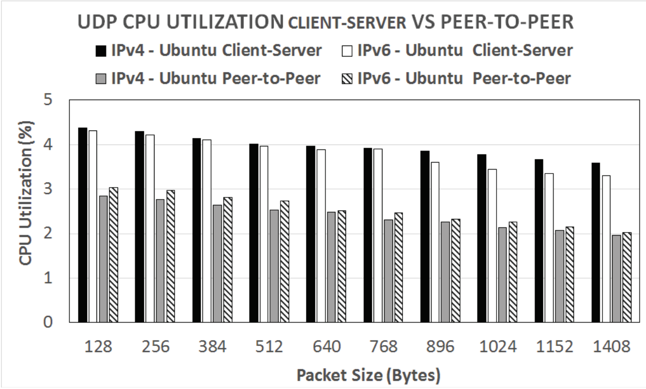

Figure 12: Ubuntu UDP CPU Utilization

Figure 12 represent UDP CPU Utilization for client-server and peer-to-peer networks. For the client-server environment, IPv6 had the highest CPU utilization of 4.37% with IPv4 being the lowest at 4.32% for packet size 128Bytes. At packet size 1408 Byte, IPv6's CPU usage was 3.59% which was higher than IPv4 by 0.29%.

For peer-to-peer, IPv6 utilization was also higher than IPv4. IPv6 had the highest CPU utilization of 3.04% at packet size 128 Bytes. At this point, IPv4's CPU usage was 2.85%. With the largest packet size, IPv6 had the highest utilization of 2.03%. For both network setups, IPv6 has the highest CPU usage with IPv4 always being the lowest. Client-server had the most CPU usage of the two network setups.

We now discuss the reason for some of the results. Due to high overhead in IPv6 packet (128 bit for addressing and more complex packet), IPv4 had higher throughput, lower latency and lower CPU utilization. UDP outperform TCP in terms of higher bandwidth, lower delay, and lower CPU utilization. This is because UDP overhead is smaller and the packet is much simpler than TCP. In addition, UDP is connectionless without error correction and does not wait to acknowledgements before send more packets. Linux overall performs better than Windows due to…. Ubuntu Linux outperformed Windows 7 and XP for IPv6 and IPv4 due to both a better kernel and how kernel network buffers are allocated and used in the Linux environment [12]. The results showed that the throughput increased with increases in packet size. This likely occurred due to the amortization of overheads associated with larger packet sizes (larger payloads)[13]. Client server is known to be faster than peer-peer as it has a dedicated server that makes it perform better, this was verified in our results. The CPU usage was higher at low packet sizes as there are many packets to be processed with small packets.

VI. Conclusion

For both network implementations studied; client-server and Peer-to-peer, our results showed that the for both Windows and Ubuntu, due to lower overhead, IPv4 performed better than IPv6, higher throughput, lower latency and lower CPU CPU utilization. Due to having a server, client server had better throughput, lower delay, but higher CPU utilization than peer-peer. Due to higher overheads, TCP had lower throughput and higher delay than UDP. Linux Ubuntu overall had better performance than Windows.

VII. Future Work

Future works will include comparing results for different channel widths in 802.11ac.

VIII. Acknowledgement

We would like to say thanks to UNITEC Institute of Technology for funding and providing equipment needed for our project. This work is part of final year's research work [14].

References

[1] Beaming, "Beamforming Wi-Fi and the 802.11ac standard," http://www.beaming.biz/support/internet-and-connectivity/beamforming-wifi-802-11ac-standard/

[2] S. S. Kolahi, N. Chand, B. K. Soorty, Z. Qu, "Performance comparison of IPv4 and IPv6 in a Peer-Peer and Client-Server Local Area Networks," IEEE 10th International Conference on Computer and Information Technology, 2010, pp. 41-45.

[3] S. S. Kolahi, H. Singla, M. N. Ehsan, and C. Dong, "the influence of WPA2 security on the UDP performance of IPv4 and IPv6 using 802.11n WLAN in Windows 7-Windows 2008 environment," Baltic Congress on Future Internet Communications, 2011, pp. 50-53.

[4] S. S. Kolahi, P. Li, "Evaluating IPv6 in Peer-to-Peer 802.11n Wireless LANs," IEEE Internet Computing, vol. 15, no.4, 2011, pp. 70-74.

[5] N. Chand, S. S. Kolahi, "TCP Throughput of IPv6 and IPv4 on IEEE 802.11n Wireless Client Server LAN in Windows Vista/XP – Server 2008 Environment," 6th International Conference on Innovative Mobile and Internet Services in IMIS, 2012, pp. 300-304.

[6] M. A. Tair, S. N. Bhatti "Introducing IEEE 802.11ac into existing WLAN deployment scenarios," IEEE 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2015, pp. 30-35.

[7] S. Narayan, C. Jayawardena, J. Wang, W. Ma, "Performance test of IEEE 802.11ac wireless devices," International Conference on Computer Communication and Informatics, 2015, pp. 1-6.

[8] S. S. Kolahi, and P. Li, P. (2016). Performance Analysis of Some Operating Systems in Wireless 802.11n Networks. International Journal of Computing and Network Technology, 4 (1), 2016, pp. 33-40.

[9] S. S. Kolahi, and A. A. Almatrook, The Impact of Human Shadowing/Movement on Performance of 802.11ac Client-to-Server WLAN. International Journal of Computing and Digital Systems (Vol. 8 (3)), 2019.

[10] Netperf, "Netperf is a benchmark that can be used to measure the performance of many different types of networking," http://www.netperf.org/netperf/NetperfPage.html

[11] D-ITG, "Distributed Internet Traffic generator," http://traffic.comics.unina.it/software/ITG/

[12] S. Zeadally, R. Wasseem, and I. Raicu, "Comparison of End-System IPv6 Protocol Stacks," IEE Proc. Communications, vol. 151, no. 3, 2004, pp. 238–242.

[13] S. Zeadally and I. Raicu, "Evaluating IPv6 on Windows and Solaris," IEEE Internet Computing, vol. 7, no. 3, 2003, pp. 51–57.

[14] A. A. Tuifaiga, P. A. Ram, Performance Analysis of IEEE 802.11ac, Unitec research report, 2019.