International Journal of Computing and Network Technology
ISSN (2210-1519)
Int. J. Com. Net. Tech. 4, No. 1 (Jan.-2016)

Performance Analysis of Some Operating Systems in Wireless 802.11n Networks

Samad S. Kolahi & Peng Li

Unitec Institute of Technology, Auckland, New Zealand

Received: 20 Oct. 2015, Revised: 20 Dec. 2015, Accepted: 22 Dec. 2015, Published: 1(January) 2016

Abstract: In this paper, the performance of some operating systems (Windows 7, Windows XP, and Fedora 12) are evaluated over open systems IEEE 802.11n WLAN (Wireless LAN), for TCP (Transmission Control Protocol) and UDP (User Datagram Protocol), IPv6 (Internet Protocol v6) and IPv4 protocols. At the time of this research, Windows 7 is still the most widely implemented operating system, although newer version of Windows operating system (Windows 8 and Windows 10) and Fedora have since been introduced but not as widely used. This paper provides better understanding of how an Operating Systems can affect link throughput and delay. We determine the throughput and delay differences between all operating systems considered. Fedora provides higher throughput and lower delay than Windows 7, while Windows XP has the lowest throughput and highest delay. Due to higher overhead, TCP provides lower bandwidth than UDP for all operating systems considered.

Keywords: Operating Systems, Wireless 802.11n, Performance Evaluation.

1. INTRODUCTION

The current generation of internet protocol, version 4 (IPv4), was introduced in 1981 [1], and is nearly 30 years old. IPv4 uses a 32-bit address space, and can provide approximately 4.3 billion unique network addresses. According to the registries that allocate Internet numbers around the world, due to the continued exponential growth of the Internet, IPv4 will run out of addresses [2]. The rapid increase in the number of Internet users also has a significant negative impact on the network performance and the quality of service (QoS) that a network can provide. In 1994, the Internet Engineering Task Force (IETF) developed a new version of Internet Protocol, first called IP next generation (IPng), and later known as Internet Protocol version 6 (IPv6) [3]. IPv6 expands the address space to 2^{128}, and provides the most up-to-date features such as real-time support, stateless address auto-configuration, QoS, security, and mobility enhancements [4]. In recent years, IPv6 have received significant attention and new versions of popular end-user operating systems have installed IPv6 by default. Consequently, hardware vendors, software developers and Internet Service Providers (ISP) are moving towards offering support for IPv6.

With the rapid advances in wireless technology and increased use of the Wireless Local Area Networks (WLAN), wireless networks have become an attractive choice for both business and personal users. The IEEE 802.11n wireless standard based products provide several benefits, including higher data rates, reliability, freedom of mobility, and backward compatibility with legacy devices. Use of Multiple-Input and Multiple-Output technology (MIMO) and doubling the channel bandwidth from 20 MHz to 40 MHz, the 802.11n theoretically supports the bandwidth of more than 200 Mbps and maximum outdoor coverage area of up to 250 meters [5] making it a threat to Fast Ethernet. IEEE 802.11n is currently the most widely used wireless standard, although newer 802.11ac is also released.

When this research started, Windows operating systems had most of the market share while Linux-based operating systems are getting more and more popular [6]. It is important to evaluate the operating systems in IPv4 and IPv6 environment in WLAN environment and see how much the operating system performance varies. At the time of this research, Windows 7 was the operating system most widely used by most companies. Using Windows 7 and Fedora, still give the comparison of performance of Linux vs Windows and how much operating system can affect the performance. In future

E-mail address: skolahi@unitec.ac.nz, penglinz@gmail.com

work, we will study the new IEEE 802.11ac standard using Windows 8, 10, and latest version of Linux operating systems.

The motivation behind this work is therefore to study some operating systems (Windows XP, Windows 7 and Fedora 12) and evaluate how much the operating system can affect the performance. and compare the results for IPv6 and IPv4 protocols using both TCP and UDP protocols. We establish test-beds and evaluate the operating systems mentioned above over 802.11n peer-peer WLAN (no encryption used, open systems).

2. Related research

Several previous research work have looked into the performance evaluation of some older operating systems using IPv4 and IPv6.

In [7], Performance of IPv4 and IPv6 using Windows XP and Windows 7 over Gigabit Ethernet wired Client-server LAN is investigated. In [8], the performance of IPv6 for TCP is evaluated for Linux and Windows over wired LAN. Authors in [9], evaluated UDP performance for some operating systems in peer-peer networks using Gigabit Ethernet. In [10], performance comparison of IPv4 and IPv6 protocol stack was conducted on some operating systems including Windows 2000 and Solaris. Their results demonstrated that IPv4 and IPv6 on Linux outperformed Windows 2000 and Solaris 8 for all the metrics used. In addition, they found out that there was a minor degradation in throughput and round-trip time (RTT) performances for IPv6 compared to IPv4 on Windows 2000 and Solaris. In [11], the IPv6 stack on different operating systems including Windows 2003, Redhat Linux 9.0 and FreeBSD 4.9 was investigated. Their study concluded that the performance of IPv6 was far better in Red Hat 9 than in Windows Server 2003. However, they did not compare their results with IPv4. In [12], the authors investigated the TCP and UDP throughput results of IPv4 and IPv6 on wireless 802.11g client-server networks using Windows XP and Vista operating systems and Server 2003 network operating system. Their results again showed that the network performance depends not only on Internet protocol version, but also on the choice of operating systems. In terms of throughput, IPv4 outperformed IPv6 on Windows client both operating systems used. In [13], the authors produced an experiment that compared the performance of Window XP, Windows Vista and Windows Server 2003 on the 802.11g networks. The authors stated that in terms of TCP throughput results, Windows XP outperformed Windows Vista by approximately 3% and Windows Server 2003 by approximately 5% on the Wireless LAN studied.

At the time this research, Windows 7 was the most widely used operating system. However, Windows 8, and Windows 10, and later version of Fedora are introduced, and these will be investigated in a future study. To the authors' knowledge, there is no research to date in literature on evaluation and comparison of these operating systems (Windows XP, Windows 7 and Fedora 12) for IPv4 and IPv6 over wireless peer-peer IEEE 802.11n LAN open systems (no security added). The main contribution of this research is therefore to obtain new results by investigating the above operating systems for IPv4 and IPv6 protocols over a peer-peer wireless LAN. Another major contribution is that the paper shows the choice of operating system affects the performance. By setting up test-beds, for both TCP and UDP protocols, throughput and RTT is measured to do the above performance evaluation.

3. Differences between IPv4 and IPv6

As discussed earlier, the main difference between IPv4 and IPv6 is in their addressing formats. IPv4 uses 32-bit address field in the IP packet header (2^{32} addresses) while IPv6 uses 128-bit addresses field (2^{128} addresses). Functions which are generally seen as working in IPv4 were kept in IPv6. IPv4 functions which are infrequently used are removed or made optional in IPv6. Other advantages of IPv6 include [3, 14]:

- Auto-configuration: IPv6 interfaces are self-configuring using IPv6 stateless auto-configuration. The system will be able to communicate with other IPv6 systems that are local and remote. In addition, it reduces the operational expenses and faults;
- Security: IPsec is mandatory in IPv6, which makes all nodes in a position to secure their traffic. IPv6 also includes security features such as payload encryption and authentication of the source of the communication in its specifications;
- Enhance QoS support: IPv6 includes labelled flows in its specifications. A flow label is defined in a specific field in the basic header, enabling the labelling and policing of traffic by the routers, without the need to inspect the application payload by the routers. This results in more efficient QoS processing;
- Mobility: IPv6 has built-in mobility which is not an add-on feature of it. Thus all IPv6 networks and nodes are IPv6-mobile ready. In addition, neighbour discovery and auto-configuration allow hosts to operate in the mobile node transparently without any specific support. IPv6 therefore is more scalable and has less redirection / re-routing (traffic optimisation) than IPv4.

A feature-by-feature comparison of IPv6 versus IPv4 is listed in Table 1:

Table 1: Comparison between IPv4 and IPv6 Features [3, 14]

Features	IPv4	IPv6
Address	32 bits long (4 bytes)	128 bits long (16 bytes)
Address Resolution Protocol (ARP)	Use to resolve an IPv4 address to the data link layer	Replaced by neighbor discovery
Address types	Unicast, multicast and broadcast	Unicast, multicast and anycast
Configuration	Manually or through DHCP (Dynamic Host Configuration Protocol)	Auto-configuration or DHCP
Fragmentation	Supported by routers and source node	Only supported by the source node
Internet Group Management Protocol (IGMP)	Used to manage local subnet group	Replaced by MLD (multicast listener discovery)
IP header	Variable length of 20-60 bytes, include checksum in header	Fixed length of 40 bytes, and no checksum in header
IP Security Protocol (IPSec)	IPSec support is optional	IPSec support is required
Quality of Services (QoS)	Differentiated services	Use traffic classes and flow labels
Mobility	Uses Mobile IPv4	Uses Mobile IPv6 with fast handover, better router optimization and hierarchical mobility

4. Experiment Setup

We set up test-beds to measure the performance of IPv4 and IPv6 on Windows XP, Windows 7 and Fedora 12. The hardware settings remained constant for all experiments conducted. Each test-bed consisted of two client machines with identical hardware comprising of an Intel® Core™ 2 Duo 6300 1.87 GHz processor with 2.00 GB RAM, Air Live Wn-5000 wireless PCI NIC card, and Western Digital Caviar 7200 (160 GB) hard-drive. The two machines were connected wirelessly (peer-peer) via Cisco Linksys WAP4410N 802.11n Access Point (AP). The test-bed diagram is displayed as Figure 1:

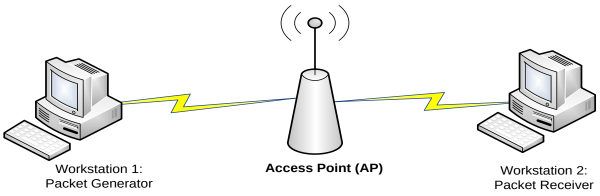

Figure 1: Network test-bed for Windows XP, Windows 7 and Fedora 12

The three different operating systems setup and configuration are explained as follows:

- In test-bed I, Microsoft Windows XP Professional with service pack 3 (SP3) is installed on both workstations. Windows XP has enabled IPv4 by default, but IPv6 is not enabled and it had to be explicitly installed and activated manually on the command line;
- In test-bed II, Microsoft Windows 7 Professional operating system is installed on both workstations. Because Windows 7 supports IPv4 and IPv6 protocol stack and supports built-in applications and services, both IPv4 and IPv6 can be enabled and configured on the two computers simultaneously by using graphic interface;
- In test-bed III, the two client machines are loaded with Fedora 12 operating system. By default, both IPv4 and IPv6 protocol stacks have been implemented in Linux kernel, thus they can be enabled and configured on the two computers simultaneously by using command lines or graphic interface.

For all of the three operating systems, the hardware is benchmarked and a similar setup is used for all the tests to negate the effect of the processor limitations and hardware design. The distance between the wireless access point and the workstations is well within two meters in order to maintain the optimum signal strength.

Parameters used for the access point configuration were:

(a) Channel bandwidth – In general, the greater the bandwidth of the assigned channels, the higher the possible speed of transmission. The access point provided two options here, 20 MHz and 40 MHz, and the latter was selected to utilize the full bandwidth.

(b) Guard Interval – The purpose of the guard interval is to introduce immunity to propagation delays, echoes and reflections, to which digital data is normally very sensitive. This function was left appropriately to its default setting on the access point.

(c) CTS (Carpal Tunnel Syndrome) Protection Mode – This function boosts the access point's ability to detect all wireless connections but severely degrades performance, hence this setting was disabled to maximize performance.

(d) Beacon Interval – This function indicates the variable times in which clients meet the access point such as sending and receiving packets, and synchronism. This setting was best left at the default interval of 100ms.

(e) DTIM (Delivery Traffic Indication Message) Interval – This setting specifies how often the access point broadcasts a Delivery Traffic Indication Message. According to the manual of the specific Linksys access point used in this project, lower settings ensure efficient networking. The default setting of 1ms therefore was left to achieve the best results.

(f) RTS Threshold – RTS (Request-to-Send) is a signal sent from the transmitting station to the receiving station requesting permission to transmit data. This setting is used to decrease the problem of the hidden stations due to distance or signal blockage [15]. The manual for the Linksys access-point recommended that this be left at the default setting of 2347 for optimum performance.

(g) Fragmentation Threshold – This specifies the number of bytes used to fragment the frames with a purpose to increase transfer reliability. If the frame size is very big, it can cause heavy interference and elevate the retransmissions rate. On the other hand, if the frame is too small, it will create overhead during the transmission and reduce the throughput rate. The parameter value for this was left at the default setting of 2346.

5. Performance Evaluation

We did the performance measurement process for both TCP and UDP in our IPv4 and IPv6 network. The performance metrics measured are the followings:

- Throughput (measured in Mbps) is the number of bits transmitted per unit time from one host to another;
- Round-trip time (measured in milliseconds) is the time required for a signal packet to travel from a specific source to a specific destination and back again.

Throughput and RTT provide a valuable insight into network performance since they are the rate at which data are transmitted from one client side to another over a network. The maximum TCP window size (64KB) and UDP window size (8KB) were used to ensure the optimum data transfer during the tests. We selected Netperf 2.4.5 [17] as the primary network traffic generation and monitoring tool to analyse the performance of IPv4 and IPv6 on the three different operating systems over an IEEE 802.11n WLAN. Netperf can determine TCP and UDP end-to-end performance across most types of networks including IPv6 network. Netperf is compatible with both Linux-based and Windows-based operating systems, and is able to measure a wide range of service parameters accurately. Netperf has been used in the past for similar research such as on the impact of wireless LAN security on performance of different Windows operating systems [12]. Zeadally et al. [10] also used Netperf for their research on end-to-end IPv6 protocol stack.

Given the load factor and the data rate, Netperf can calculate the theoretical maximum rate at which the network link should be able to process the data. In this research, we used various IP packet sizes ranging from 128 to 1408 bytes which cater for most packet sizes on networks and the Internet [18]. Most performance evaluation tests were executed for a period of about 60 seconds, which usually generated one million packets of a particular packet size and protocol (one run). To ensure accuracy of the result and rule out any inconsistencies, we repeated each tests for 40 runs and the results then average and standard deviation of results was calculated.

6. Experimental Results

The experiments were performed and throughput and round-trip were measured for both TCP and UDP protocol in an IEEE 802.11n network. Data packet sizes were gradually increased in size from 128 to 1408 bytes for both TCP and UDP and the resulting throughput and RTT values were plotted. This was done for Windows XP, Windows 7 and Linux Fedora 12 operating systems.

A. Throughput

TCP and UDP throughput results for the three operating systems used and for IPv4 and IPv6 protocols for the different packet sizes are illustrated in Figures 2 and 3.

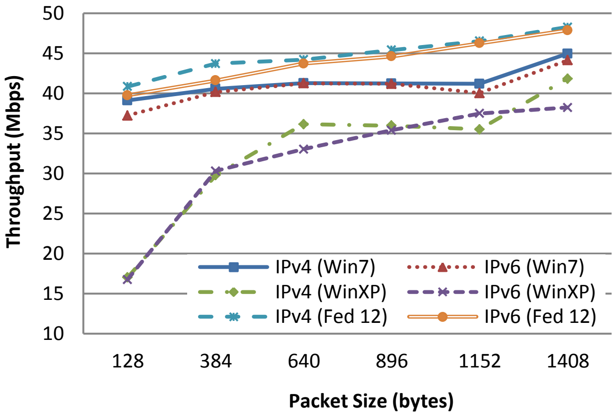

Figure 2: TCP Throughput of Windows and Fedora 12 Operating Systems for 802.11n (open systems)

From the TCP throughput results in Figure 2, we observe that for all the three operating systems there were performance differences between IPv4 and IPv6 throughput. However, the variations in TCP throughput values were different for various operating systems. For

Windows 7, IPv4 outperformed IPv6 for all packet sizes. IPv4 TCP throughput results ranged from 39.12 to 44.96 Mbps, and IPv6 TCP throughput values were from 37.24 to 44.13 Mbps. The maximum difference between IPv4 and IPv6 for Windows 7 was 1.88 Mbps on packet size 128 bytes. Similarly, on Windows XP IPv4 had higher TCP throughput than IPv6 for the most packet sizes except for the packet size 1152 bytes, where IPv6 provided 1.98 Mbps (37.50 Mbps compared to 35.52 Mbps) more TCP throughput than IPv4. The highest point of difference in Windows XP between IPv4 and IPv6 was 3.11 Mbps on packet size 640 bytes. Fedora 12 throughputs exhibited that IPv4 performed better than IPv6 for all packet sizes. The highest gap between IPv4 and IPv6 for Fedora 12 was 2.10 Mbps on packet size of 384 bytes.

Figure 2 also shows that among the operating systems, the highest TCP bandwidths were for Fedora 12 (48.27 Mbps for IPv4 and 47.88 Mbps for IPv6 at packet size of 1408), followed by Windows 7 (44.96 Mbps for IPv4 and 44.13 for IPv6) and the lowest bandwidth were for Windows XP (41.83 Mbps for IPv4 and 38.23 for IPv6).

The standard deviation for the above throughput results are recorded in the Table 2:

Table 2: Standard Deviation for TCP Throughput

acket size (Bytes)	Windows 7		Windows XP		Fedora 12	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
128	0.93	0.91	0.55	0.98	0.97	0.85
384	0.61	0.88	0.98	0.99	0.65	0.93
640	0.97	0.88	0.87	0.88	0.96	0.64
896	0.82	1.00	0.93	0.96	0.96	0.97
1152	0.88	1.28	0.92	0.99	0.98	0.58
1408	0.96	1.07	0.90	0.98	0.98	0.83

UDP throughput results for IPv4 and IPv6 over Windows 7, Windows XP and Fedora 12 with packet sizes ranging from 128 to 1408 bytes is illustrated in Figure 3.

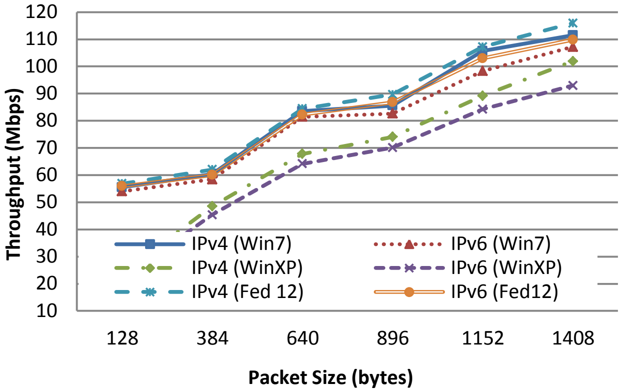

Figure 3: UDP Throughput of Windows and Fedora 12 Operating Systems for 802.11n (open systems)

As Figure 3 indicates, the UDP throughput results we obtained were much higher than the TCP throughput values. For all operating systems, the highest TCP throughput was 48.27 Mbps (Fedora 12 using IPv4) while the highest UDP throughput was 115.92 Mbps (Fedora 12 using IPv4). This is because UDP is a connectionless protocol and the source does not have to wait for acknowledgements since the destination does not send any acknowledgements. These results is in contrast with the cable LAN results in [10] where TCP and UDP results were close and had up to 10 Mbps difference (approximately 10% difference). This difference between cable LAN results in [10] and wireless LAN results of this study could be possibly because of the CSMA/CA (carrier sense multiple access/ collision avoidance) media access control used on wireless LAN where TCP Acknowledgements have more effect in wireless LAN than cable LAN (UDP send back no acknowledgments as stated above). However, the results show that at low packet sizes the difference between TCP and UDP was less significant.

IPv4 again outperformed IPv6 for all operating systems. For Windows 7, IPv4 provided higher UDP throughput than IPv6 for all of the packet sizes. The maximum difference was noticeable at packet size 1152 bytes, where IPv4 outperformed IPv6 by 7.37 Mbps. For Windows XP, the most noticeable difference of UDP throughput between IPv4 and IPv6 was 8.98 Mbps on packet size 1408 bytes.

Also as shown in Figure 3, Fedora 12 had the highest throughput for both IPv4 and IPv6 for all packet sizes. For Fedora 12, the highest UDP throughputs were 115.92 Mbps (for IPv4) and 109.78 Mbps (for IPv6) for packet size of 1408. The next highest throughputs were for Windows 7 at 111.46 Mbps (for IPv4) and 107.21 Mbps (for IPv6) and finally Windows XP had 101.93 Mbps (for IPv4) and 92.95 Mbps for IPv6.

The standard deviation for the above results is shown in Table 3:

Table 3: Standard Deviation for UDP Throughput

Packet size (Bytes)	Windows 7		Windows XP		Fedora 12	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
128	0.93	0.91	0.98	0.86	0.98	0.73
384	0.61	0.88	0.99	0.96	0.47	0.48
640	0.97	0.88	0.88	0.97	0.52	0.65
896	0.82	1.00	0.96	0.96	0.68	0.94
1152	0.88	1.28	0.99	0.91	0.91	0.93
1408	0.96	1.07	0.98	0.98	0.98	0.87

The gain in TCP and UDP throughput as packet size increase is likely to the amortization of overheads associated with larger user packet sizes (larger user payloads) [7].

B. Round-trip time

Round-Trip Time is also an important performance metric. The TCP and UDP round-trip time results are shown in Figures 4 and 5.

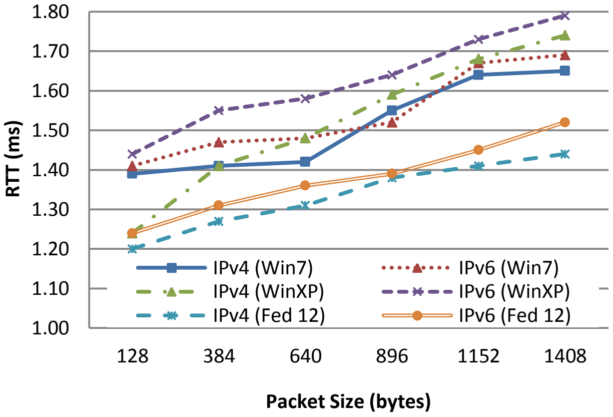

Figure 4: TCP RTT of Windows and Fedora 12 Operating Systems for 802.11n (open systems)

As shown in Figure 4, for all the three operating systems, IPv4 outperformed IPv6 for all packet sizes. On Windows 7, the highest difference of TCP RTT between IPv4 and IPv6 was 0.06 ms on packet sizes 384 and 640 bytes. Likewise, on Windows XP, the maximum difference was at packet size 384 bytes, where IPv4 had 0.14 ms less TCP RTT values than IPv6. For Fedora 12, the highest point of difference between IPv4 and IPv6 can be noted at the packet size 1408 bytes, where IPv4 had 0.08ms lower TCP latency than IPv6.

Comparing the TCP RTT of the three operating systems, Fedora 12 had the least TCP RTT results for both IPv4 and IPv6 for all packet sizes while Windows XP had the highest RTT values. The highest gap between Windows XP and Fedora 12 was noticed at packet sizes 1408 bytes for IPv4 and 1152 bytes for IPv6; where IPv4 on Windows XP had 0.3ms higher TCP RTT and IPv6 had 0.28ms more TCP RTT than Fedora 12. The maximum difference between Windows 7 and Fedora 12 was noticed at packet size of 1152 bytes for both IPv4 and IPv6, where IPv4 on Fedora 12 had 0.23ms less TCP RTT, and IPv6 had 0.22ms lower RTT values than Windows 7. We also found out that Windows 7 had lower TCP RTT than Windows XP for all packet sizes, with one exception at packet size 128 bytes for IPv4, where Windows 7 had 0.15ms higher TCP RTT.

The standard deviation of TCP round-trip time results is shown in Table 4.

Table 4: Standard Deviation for TCP RTT

Packet size (Bytes)	Windows 7		Windows XP		Fedora 12	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
128	0.09	0.03	0.02	0.02	0.02	0.03
384	0.05	0.03	0.07	0.04	0.06	0.02
640	0.02	0.03	0.02	0.01	0.05	0.03
896	0.03	0.03	0.06	0.01	0.04	0.02
1152	0.03	0.04	0.05	0.01	0.03	0.02
1408	0.02	0.02	0.02	0.04	0.02	0.02

The UDP RTT results depicted in Figure 5, also show that IPv6 had slightly higher delay than IPv4 on all the three operating systems.

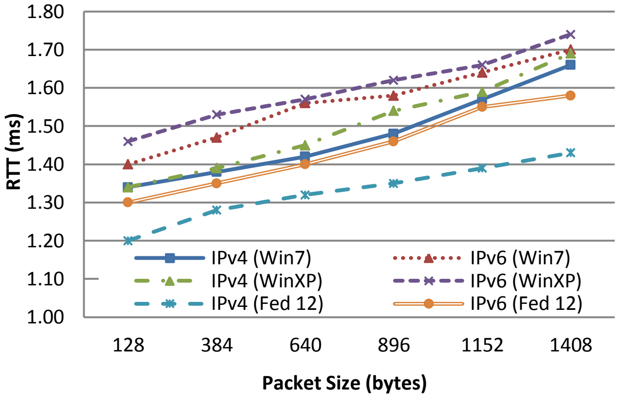

Figure 5: UDP RTT of Windows and Fedora 12 Operating Systems for 802.11n (open systems)

Figure 5 shows that on Windows 7, IPv4 had better performance than IPv6 for all packet sizes. The greatest difference between IPv4 and IPv6 was about 0.14 ms on packet size 640 bytes. In the case of Windows XP, the highest gap was on packet size 1384 bytes, where IPv4 had 0.14 ms less UDP RTT than IPv6. For Fedora 12, the most noticeable difference of UDP RTT between IPv4 and IPv6 was 0.16ms on packet size 1152 bytes.

Fedora 12 had better UDP RTT performance than the two Windows-based operating systems. The maximum gap between Fedora 12 and Windows 7 was at packet size 1408 bytes for both IPv4 and IPv6, where Window 7 had 0.23ms more UDP RTT values on IPv4 and 0.12ms higher UDP RTT values on IPv6 than Fedora 12. Comparing the UDP RTT results of Fedora 12 with Windows XP results, Fedora 12 had lower UDP RTT than Windows XP for all packet sizes for both IPv4 and IPv6. The highest point of difference between the two operating systems was at packet size 1408 bytes for IPv4 and 384 bytes for IPv6, where Windows XP had 0.26ms more UDP RTT for IPv4 and 0.18ms higher UDP RTT for IPv6 than Fedora 12. The difference between Windows 7 and Windows XP was less significant and most noticeable at packet size 896 bytes for IPv4 and 384 bytes for IPv6, where Windows 7 had 0.06ms less UDP RTT for both IPv4 and IPv6 than Windows XP.

The standard deviation of UDP round-trip time results is shown in Table 5.

Table 5: Standard Deviation for UDP RTT

Packet size (Bytes)	Windows 7		Windows XP		Fedora 12	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
128	0.03	0.03	0.03	0.02	0.04	0.03
384	0.05	0.04	0.03	0.03	0.04	0.02
640	0.07	0.05	0.02	0.03	0.03	0.03
896	0.03	0.03	0.04	0.03	0.02	0.03
1152	0.05	0.03	0.04	0.03	0.01	0.05
1408	0.07	0.06	0.04	0.04	0.03	0.03

As can be seen from Figures 2 to 5, IPv4 performed better than IPv6 for both TCP and UDP throughput and RTT on all of the three operating systems. This difference is likely due to the large IPv6 header size of 40 bytes compared to IPv4's header size of 20 bytes which affects the performance of IPv6. Linux-based operating system, Fedora 12, outperformed Windows 7 and Windows XP for both TCP and UDP protocols. Researchers in [11] also observed that Linux performs better in IPv6 environment than Windows. This is probably because of the way kernel network buffers are allocated and used by Linux operating systems. Linux platforms are based on the traditional BSD (Berkeley Software Distribution) which have pre-allocation of a number of fixed-sized memory buffers. When a network application transmits data, the pre-allocated buffers are used to avoid overheads associated with buffer allocations [19, 20].

Comparing the Windows 7 with Windows XP, it was obvious that Windows 7 had higher TCP and UDP bandwidth and lower RTT for most packet sizes on both IPv4 and IPv6. In the absence of the operating system source codes, we may, therefore, reasonably conclude that Microsoft has integrated changes in newer kernels for Windows 7 to improve its overall performance.

7. Conclusions

In this paper, we carried out several experimental performance comparisons between the IPv4 and IPv6 protocol stack on the operating systems including Windows XP, Windows 7 and Fedora 12 over Peer-to-Peer 802.11n WLAN with no security added. At the time of this research, Windows 7 was the operating system most widely used by most companies. Using TCP and UDP throughput and round-trip time as the metric, our experimental results show that, for all the three operating systems, IPv4 outperformed IPv6 on most packet sizes for both TCP and UDP traffic. Although IPv6 has many advantages, it also has its drawbacks by having lower bandwidth and higher delay compared to IPv4 due to IPv6 larger header size.

For both IPv4 and IPv6, Fedora 12 had better TCP and UDP traffic performance than Windows 7 and XP. The newer Windows operating system, Windows 7, had improved IPv6 performance than Windows XP.

8. Future work

In future, we plan to extent this study by incorporating more operating systems and a greater range of metrics. We will study the new 802.11ac WLAN standard using Windows 8 or 10 and later versions on Linux. In addition, the performance comparison of Windows and Red hat Linux platforms with IPv4 and IPv6 using wired LAN on 64-bit operating system will be investigated. Future work also includes evaluating performance of inter-operating technologies between IPv4 and IPv6 such as 6to4 protocol.

Acknowledgment

The authors would like to thank UNITEC Institute of Technology for helping the research team and providing the inventory needed.

1. REFERENCES

[1]. Information Sciences Institute USC: "Internet Protocol," RFC 791, 1981.

[2]. OECD report, Internet Address Space: Economic Considerations in the Management of IPv4 and in the Deployment of IPv6, OECD Ministerial Meeting, Seoul, Korea, pp. 17-18, 2008.

[3]. Deering, S., and Hinden, R..: 'Internet protocol, version 6 (IPv6) specification', RFC 1883, Internet Engineering Task Force, December 1995.

[4]. D.Lee et al. 'The Next Generation of the Internet: Aspects of the Internet Protocol Version 6', IEEE Network, vol. 12, no. 1, pp. 28-33, 1998.

[5]. IEEE P802.11n/D1.0 Draft Amendment to STANDARD[FOR] Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-Specific requirements-Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications: Enhancements Mar 2006.

[6]. NetApplications,http://marketshare.hitslink.com/operating-system market-share.aspx?qprid=8, accessed October 2010.

[7]. Jain, P., Singh, S., Singh, G., and Goel, C. "Performance Comparison of IPv4 and IPv6 using Windows XP and Windows 7 over Gigabit Ethernet LAN." *International Journal of Computer Applications (0975 – 8887), 43*(16), 2012.

[8]. Soorty,B, Sarkar, N., "Quantifying the performance degradation of IPv6 for TCP in windows and Linux networking", Australasian Telecommunication Networks and Applications Conference (ATNAC), pp 25-29, 2013.

[9]. Soorty, B.; Sarkar, N.I., "Evaluating IPv6 in peer-to-peer Gigabit Ethernet for UDP using modern operating systems**",** IEEE Symposium on Computers and Communications (ISCC), pp.: 534 -536, 2012.

[10]. Zeadally, S., and Raicu, L.: 'Evaluating IPv6 on Windows and Solaris', Internet Computing, IEEE, vol. 7, no. 3, pp. 51-57, 2003.

[11]. Mohamed, S., Abusin, A., Chieng. D.: 'Evaluation of IPv6 and Comparative Study with Different Operating Systems', Information Technology and Applications, vol. 2, pp. 665-670, 2005.

[12]. Kolahi, S. S., Narayan, S., Sunarto, Y., Mani, P.: 'Performance Comparison of IPv4 and IPv6 on Various Windows Operating Systems', The 11th International Conference on Computer and Information, Technology (ICCIT 2008), pp. 663-668, 2008.

[13]. Kolahi, S. S., Qu, Z., Soorty, B., Chand, N.: 'The Impact of Security on the Performance of IPv4 and IPv6 using 802.11n wireless LAN', The 3rd International Conference on New Technologies, Mobility and Security (NTMS 2009), pp. 1-4, 2009.

[14]. Tanenbaum, A.: 'Computer Networks', Prentice Hall INC, New Jercy, 3rd ed, 1996

[15]. Akin, D., and Geier, J.: '802.11 PHY layers', CWAP - certified wireless analysis professional official study guide, Mc.Graw-Hill, pp. 353-355, 2004.

[16]. Konstantinos, A., Sotiros, K.G., John, N.S.: 'A Test Lab for the Performance Analysis of TCP over Ethernet LAN on Windows Operating Systems', IEEE Transaction on Educcation, 48, (2), pp.318-328, 2005.

[17]. Jones, R.: 'Netperf'. http://www.netperf.org/netperf/NetperfNew.html, accessed March 2010.

[18]. Thompson, K., Miller, G. J., Wilder, R.: 'Wide-area Internet traffic patterns and characteristics', IEEE Network, Volume 11, Issue 6, pp. 10-23, Nov-Dec 1997.

[19]. Zeadally, S., Wasseem, R., Raicu, L.: 'Comparison of end-system IPv6 protocol stacks,' IEE Proceedings Communications, pp. 238-242, 2004.

[20]. Murugesan, R., Ramadass, S., Budiarto, R.: 'Improving the Performance of IPv6 Packet Transmission over LAN'. 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA), pp. 182-187, 2009.

Samad Kolahi receved his PhD from Auckalnd University, Auckland, New Zealand, and curently is a senior lecturer at Unitec Institute of Technology, New Zealand. He has ten years experince in telecommunication industry as a network designer and principal planning engineer. Samad is a senior member of the IEEE and a committee member of the IEEE New Zealand. His reserch interrest are local aeaa network evaluation, cyber security, and cellular newtroks.

Peng Li received his BCS in computer science from the Depaertment of Computing and Information Technology, Unitec Institute of Technology, New Zealand. His research interests include wirelss LAN, and Perfroamnce Evaluation of LANs.