

Evaluating IPv6 in Peer-to-Peer 802.11n Wireless LAN

Samad S. Kolahi, Peng Li

Unitec New Zealand, New Zealand

Internet Protocol version 6 (IPv6) is built into the latest versions of Microsoft Windows and Linux based operating systems. IPv6 is expected to replace the current IPv4 in near future and is designed to solve its numerous problems, such as address exhaustion, security, and mobility. It is important to know what would be the drawbacks of implementing IPv6 and determine which of the modern operating systems (Windows XP, Windows 7 and Fedora 12) would give the best performance over IPv6 networks. Our results indicate that IPv6 performs overall better in Fedora 12 environment and that IPv6, due to larger overhead, has lower bandwidth and higher round trip time than IPv4.

Internet is becoming the most powerful communication channel for enterprises and home users. However, the current IPv4 will run out of addresses in the near future.¹ IPv6 has been selected as the successor of IPv4, and designed not only to increase the address space, but also to consist of the most up-to-date features, such as real-time audio and video delivery, stateless auto-configuration, security, quality of service (QoS), and mobility enhancements.² IPv6 can provide (2^{128}) addresses while IPv4 provides only (2^{32}) addresses (128 bits address space for IPv6 versus 32 bits for IPv4). The drawback of the large address size is that IPv6 has a larger overhead in the packet (40-byte header while IPv4 has a 20-byte header³). This increase in header size has some implications for the performance of the IPv6.

With IPv6 being implemented and with the growth of wireless IEEE 802.11n usage, and with the release of Windows 7 operating system, it is important to investigate the performance of IPv6 networks with such latest developments.

The motivation for this study is therefore to study the implementation of IPv6 and compare it with IPv4 using the latest and most popular operating systems over a wireless peer-peer IEEE 802.11n local area networks. Such results will help industry professionals and researchers to know what to expect when implanting IPv6 using Windows XP, Windows 7, and Fedora 12. Our experimental results confirm the previous studies³ (on older operating systems) that various operating systems affect the performance of IPv6.

Related Work in IPv6 Performance

Performance evaluation of IPv6 on different operating systems has been conducted by a number of previous researchers. In 2003, Zeadally et al. carried out an empirical performance comparison of IPv4 and IPv6 protocol stack implementations on Windows 2000, Solaris and Linux operating systems.³ They used a number of metrics and evaluated IPv6's performance compared to IPv4 on Windows 2000 and Solaris. In 2004, Zeadally et al. conducted another experimental performance evaluation of IPv4 and IPv6 on Solaris 8 and Windows 2000.⁴ From the study, the researchers found that Solaris 8 gives better performance in IPv6 environment.

In 2005, Mohamed et al. investigated the IPv6 protocol stack on different operating systems including Windows 2003, Redhat Linux 9.0 and FreeBSD4.9 over a wired LAN.⁵ They used various network metrics including throughput and Round-Trip Time (RTT) to evaluate the performance of IPv6 with the different packet sizes. The researchers did not compare their results with IPv4.

In 2009 Kolahi et al. conducted a wireless 802.11n study (with and without wireless security) on the performance comparison of IPv4 and IPv6 on client-server networks, using Windows XP, Vista, and Windows Server 2008.⁶

The operating systems or protocols that were implemented in the above studies are mostly out of date now, and therefore new studies are required on latest operating systems using the latest wireless protocol. To the authors' knowledge, there is no work done to-date in literature on evaluation of IPv6 using IEEE

802.11n peer-peer LAN and various modern operating systems, Windows XP, Windows 7 and Fedora 12. The contribution of this research is to obtain new results by investigating the drawbacks of implementing IPv6 (in terms of delay and bandwidth) and further investigate which of the commonly used operating systems would give the best performance for IPv6.

Test Bed and Measurement Procedure

To evaluate the performance of IPv6 and to compare the results with IPv4 on Windows XP, Windows 7 and Fedora 12, we implemented a wireless peer-to-peer IEEE 802.11n network with WPA2 security, and measured throughput, Round-Trip Time and CPU utilization over IPv4 and IPv6 protocol stacks.

Test Bed Configuration

We connected two client machines with identical hardware (CPU: Intel® Core™ 2 Duo 6300 1.87 GHz, RAM: 2.00 GB, Network card: Air Live Wn-5000 wireless PCI NIC, Hard drive: Western Digital Caviar 7200 [160 GB]) wirelessly via Cisco Linksys WAP4410N 802.11n Access Point (AP). The test-bed setup remained constant for all experiments conducted.

According to Killelea⁷, there are several factors which will affect the performance of a network, such as process limitations and hardware designs. In-order to eliminate the effect of such factors, the hardware is benchmarked and the same setup is used for all the tests. The distance between the access point and the workstations was well within two meters to maintain the optimum signal strength. We selected 40 MHz for the access points channel bandwidth to utilize the full bandwidth, and chose WPA2 security encryption techniques to secure the wireless network.

In three test-beds, Microsoft Windows XP Professional with SP3 (service pack 3), Microsoft Windows 7, and Linux Fedora 12 operating systems were installed on both client machines. The above set-ups were done separately for IPv4 and IPv6 and then we compared the results.

Measurement Procedure

We selected Netperf 2.4.5 as the primary network traffic generation and monitoring tool (www.netperf.org/netperf/NetperfNew.html) to measure the data transfer and analyze the performance of IPv6 on the three operating systems over IEEE 802.11n WLAN. Netperf has been used in the past for similar research, such as the impact of wireless LAN security on performance of different Windows operating systems.⁶

In order to evaluate the performance of IPv6 over IEEE 802.11n wireless peer-peer LAN using common operating systems, the metrics used in our tests were throughput (measured in Mbps), RTT (round-trip time, measured in milliseconds), and CPU utilization (measured as a percentage).

Most performance evaluation tests were executed for a period of about 60 seconds, and that usually generated one million packets (one run). To ensure accuracy of the result and

rule out any inconsistencies, we repeated each test until 95% confidence interval was achieved (this was achieved after approximately 40 runs.)

Experiment Results

For both IPv6 and IPv4 protocols, we measured the performance parameters mentioned above for both TCP and UDP protocols. Data packet sizes is gradually increased (128, 384, 640, 896, 1152, and 1408 bytes) as the packet size impacts the results.

IPv6 vs IPv4 Throughput

Throughput is one of the measures of the comparative effectiveness of a system in a given time period. As shown in Figure 1, for 802.11n wireless peer-peer LAN with WPA2 security, the TCP throughput of IPv4 was higher than the TCP throughput values of IPv6 for all operating systems and for most packet sizes. The maximum difference between IPv4 and IPv6 was 3.56 Mbps (32.80 Mbps for IPv4 compared to 29.24 Mbps for IPv6) and that was for Windows XP at packet size of 640 bytes.

For Fedora 12, the highest point of difference between IPv4 and IPv6 can be noted at the packet size of 1152 bytes where IPv4 provided a 980 Kbps higher throughput value than IPv6 (45.54 Mbps for IPv4 compared to 44.56 Mbps for IPv6). On Windows 7, the highest gap between IPv4 and IPv6 was on packet size of 1152 bytes where IPv4 provided 2.63 Mbps higher TCP throughput than IPv6 (40.72 Mbps for IPv4 compared to 38.09 Mbps for IPv6).

Comparing the three operating systems, as shown in Figure 1, for both IPv6 and IPv4, Fedora 12 recorded the highest TCP throughput. Windows 7 gave the next highest TCP throughput, while Windows XP had the lowest throughput value. For IPv6, the highest TCP throughput was 45.14 Mbps (Fedora 12) followed by 41.30 Mbps (Windows 7) and 35.72 Mbps (Windows XP). In IPv4, the highest bandwidths were 46.11 Mbps (Fedora 12), 41.30 Mbps (Windows 7) and 35.72 Mbps (Windows XP). Fedora 12 outperformed Windows 7 and XP for IPv6 and IPv4 because Fedora 12 operating system has a better kernel and the way kernel network buffers are allocated and used in Linux environment⁴. Linux has 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. Kernel is the most crucial part of most computer operating systems and is responsible to manage the resources of the system between the hardware and the software.

As Figure 1 indicates, Peer-to-Peer wireless LAN with WPA2 security can provide up to 46.1Mbps TCP throughput. This TCP throughput is much less than what we achieved (180 Mbps) in IEEE 802.11n wireless client-server networks experiments when one of the links (to the server) was cable while in peer-peer wireless LAN both of the links to the access points were wireless. Our experiments further showed that WPA2 security can reduce the peer-peer bandwidth by up to 4.5 Mbps due to WPA2 overheads.

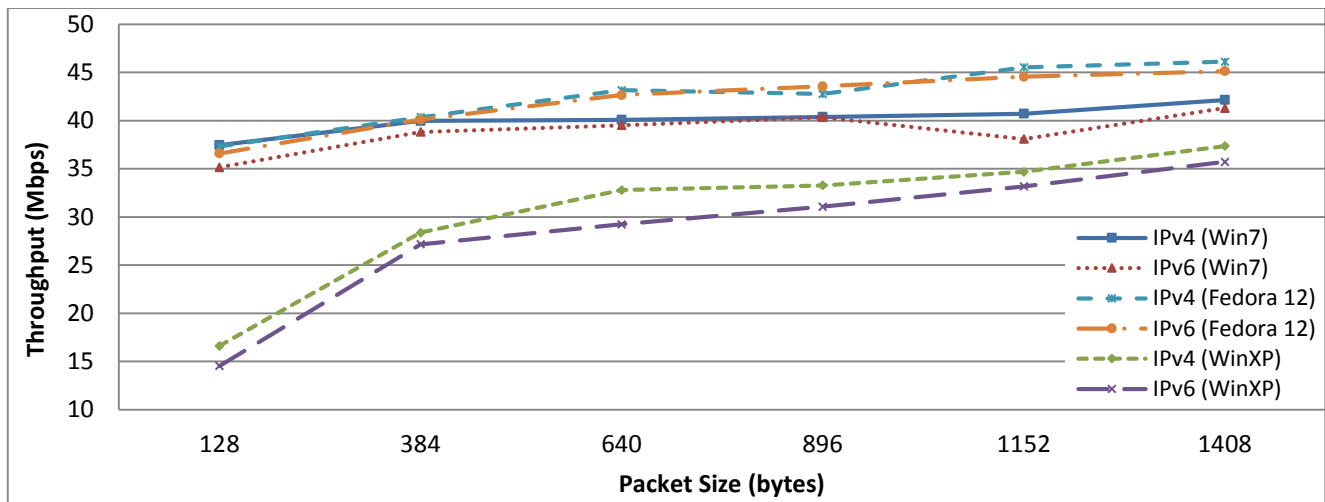


Figure 1. TCP throughput for IPv4 and IPv6 over Windows XP, Windows 7 and Fedora 12.

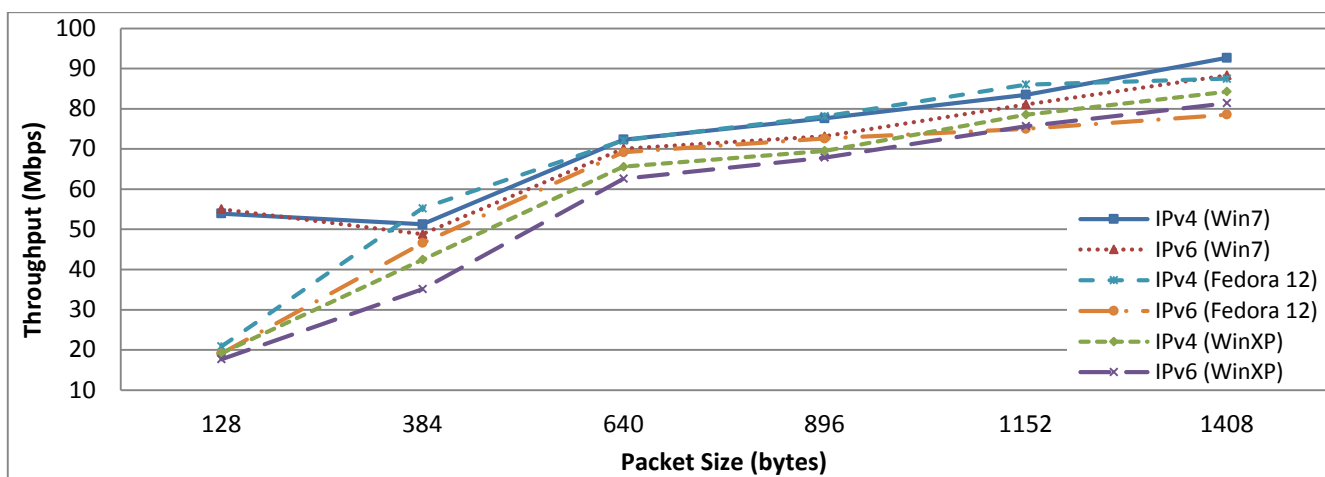


Figure 2. UDP throughput for IPv4 and IPv6 over Windows XP, Windows 7 and Fedora 12.

Figure 2 indicates that IPv4 has higher UDP throughput than IPv6 for all operating systems investigated and for most packet sizes. Among the three operating systems, Fedora 12 had the highest difference between IPv4 and IPv6 throughput performance as IPv4 had 11.03 Mbps more bandwidth than IPv6 (86.02 Mbps for IPv4 compared to 74.99 Mbps for IPv6) on packet size of 1152 bytes. Also on Windows 7, for most of packet sizes, IPv4 provided higher UDP throughput than IPv6. The maximum difference was noticeable at packet size of 896 bytes, where IPv4 outperformed IPv6 by 4.46 Mbps (77.60 Mbps for IPv4 compared to 73.14 Mbps for IPv6). For Windows 7 and packet size of 128 bytes, we had unexpectedly high throughput. Results of some of the runs for 128 byte packets were inconsistent and inconclusive. For Windows XP, the most noticeable difference of UDP throughput between IPv4 and IPv6 was 7.35 Mbps (42.45 Mbps for IPv4 compared to 35.10 Mbps for IPv6) on packet size of 384 bytes.

The highest maximum UDP throughput for Windows 7 was 92.65 Mbps for IPv4 and 88.30 Mbps for IPv6 at packet size of 1408 bytes. Fedora 12 maximum throughputs were 87.46 Mbps for IPv4 and 78.53 Mbps for IPv6. For Windows XP the

maximums were 84.25 Mbps for IPv4 and 81.30 Mbps for IPv6. For most packet sizes, Fedora 12 had higher throughput than Windows XP for IPv6 and IPv4. For packet sizes of 128 and 1152 bytes, the difference between Fedora 12 and XP was less significant.

UDP throughput values were much higher than the TCP throughput results. This is because UDP is a connectionless protocol and does not have to wait for acknowledgements as it does not send any acknowledgements. These results are in contrast with the cable LAN results in³ where TCP and UDP results were close and had up to 10 Mbps difference (approximately 10% difference). In our wireless peer-peer LAN experiments, we observed up to 45 Mbps difference (100% difference) between TCP and UDP. TCP bandwidth varied between 15 to 47 Mbps while UDP bandwidth was between 20 and 93 Mbps. This difference between cable LAN results in³ 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 in 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.

The lower throughput results for IPv6 than IPv4, for all operating systems considered and for both TCP and UDP, is due to the 40-byte header of IPv6 compared to IPv4 20-byte header^{3,8} which had implications for the performance of the IPv6, resulting in lower bandwidth for IPv6.

IPv6 vs IPv4 Round-Trip Time

Round-Trip Time as a measure of latency is also an important performance metric. Figure 3 shows that IPv6 has higher TCP RTT than IPv4 for all operating systems investigated with the maximum difference of 0.24ms (1.88ms for IPv6 compared to 1.64ms for IPv4) observed for Windows XP and packet size of 1408 bytes. On Windows 7 the maximum difference was at packet size of 896 bytes, where IPv6 had 0.11ms higher RTT than IPv4 (1.71ms for IPv6 compared to 1.60ms for IPv4). On Fedora 12, the highest point of difference between IPv4 and IPv6 can be noted at the packet size of 640 bytes where IPv6 provided a higher RTT of 0.15ms than IPv4 (1.56ms for IPv6 compared to 1.41ms for IPv4).

Among the three operating systems considered, Windows XP with IPv6 had the highest TCP RTT than the other operating systems for most packet sizes. Fedora 12 had the lowest TCP RTT and outperformed Windows 7 and Windows XP for both IPv4 and IPv6.

The UDP RTT results depicted in Figure 4 also showed that IPv6 had slightly higher RTT than IPv4 on Windows 7 and Windows XP. In Windows 7 and for all packet sizes, the differences between IPv4 and IPv6 were approximately between 0.02ms-0.03ms (for example 1.75ms for IPv4 compared to 1.78ms for IPv6 at packet size of 1408 bytes). For Windows XP, the maximum difference was 0.13ms at packet size of 384 bytes (1.42ms for IPv4 compared to 1.55ms for IPv6). Among the three operating systems, Windows XP had the highest difference in RTT between IPv6 and IPv4 (up to 0.13ms difference). For Fedora 12, IPv4 had higher RTT than IPv6 on 640, 896, 1152, and 1408 bytes packet sizes, while IPv6 had higher RTT for 128 and 384 bytes packet sizes. The highest gap was on packet size of 1408 bytes where IPv4 had 0.10ms higher RTT than IPv6 (1.73ms for IPv4 compared to 1.63ms for IPv6).

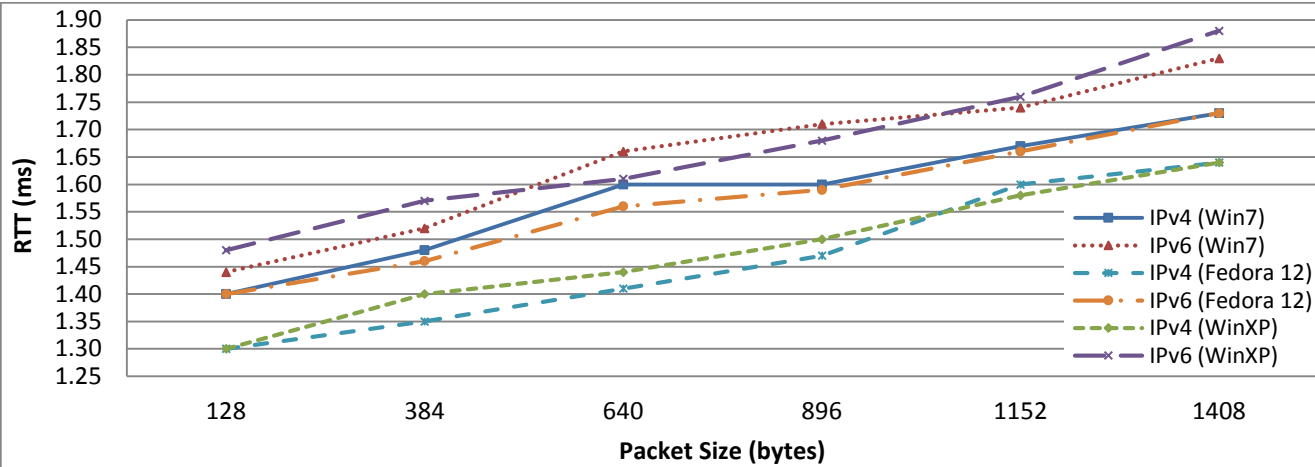


Figure 3. TCP RTT for IPv4 and IPv6 over Windows XP, Windows 7 and Fedora 12.

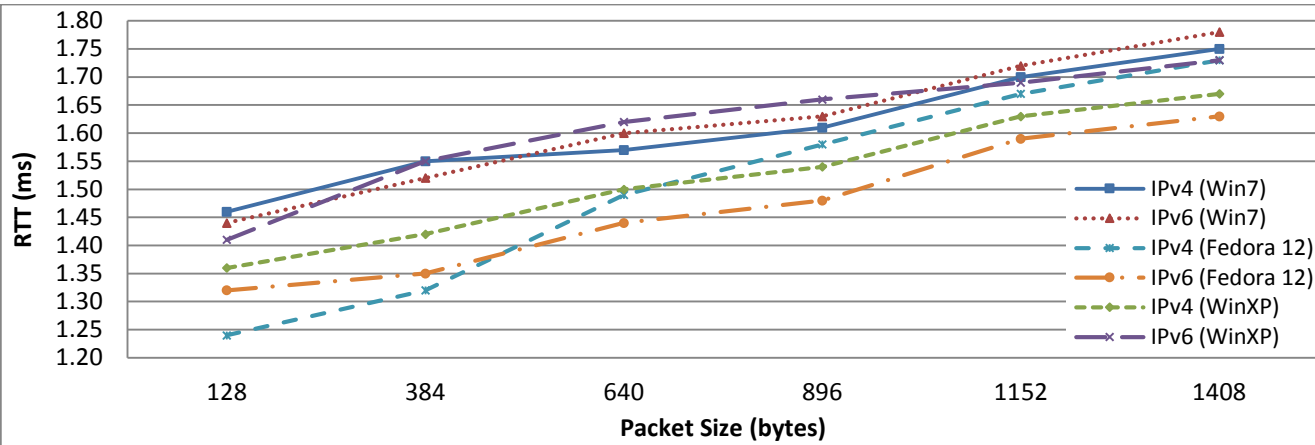


Figure 4. UDP RTT for IPv4 and IPv6 over Windows XP, Windows 7 and Fedora 12.

As we see in Figure 4, for IPv6, Fedora 12 had better UDP RTT performance than the two Windows-based operating systems. The IPv6 RTT ranged from 1.32ms (for packet size of 128 bytes) to 1.63ms (for packet size of 1408 bytes) for Fedora 12, between 1.41ms to 1.73 for Windows XP and between 1.44ms to 1.78 for Windows 7. For IPv4, Fedora again has lower RTT than the other two operating systems. The exception was that Fedora 12 had marginally higher UDP RTT than XP for the packet sizes of 896 to 1408 bytes. The IPv4 RTT range was 1.28ms to 1.73ms for Fedora 12, 1.46ms to 1.75ms for Windows 7, and between 1.36ms to 1.67ms for Windows XP. It is interesting to note that the increase in RTT with the increase in each packet size. It is likely due to the amortization of overheads associated with larger packet sizes (larger payloads)³.

CPU utilisation is an important resource to be managed in order to run the operating systems efficiently. We measured CPU usage at sending node during our throughput experiments (Figure 5). Fedora 12 (ranging from 25% to 47%) used more CPU resources than Windows 7 (ranging from 15% to 37%) and Windows XP (ranging from 13% to 30%) for both IPv4 and IPv6. The CPU usage was higher at small packet sizes, and it dropped off steadily as the packet size increased. It also can be observed from Figure 5 that IPv6 utilized more CPU resources than IPv4 for all the three operating systems considered, but the range was within 5% for all packet sizes. This is much less than 20% difference observed for Windows 2000 and Solaris reported in³. As the higher CPU utilization of a process corresponds to a higher load on the system, IPv6 therefore had more processing load due to higher features and higher overhead.

CPU Utilization

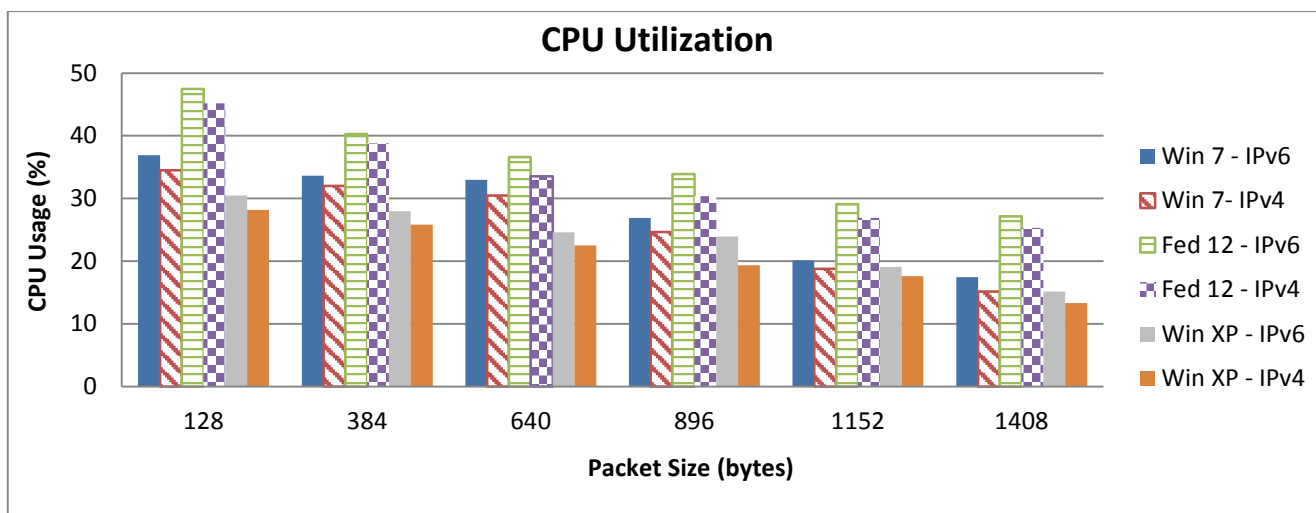


Figure 5. CPU Utilization for IPv6 and IPv4 on Windows XP, Windows 7 and Fedora 12.

References

1. OECD report, Internet Address Space: Economic Considerations in the Management of IPv4 and in the Deployment of IPv6, OECD Ministerial Meeting, Seoul, Korea, 17-18 June 2008.
2. 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.
3. S. Zeadally, and L. Raicu, "Evaluating IPv6 on Windows and Solaris," *IEEE Internet Computing*, vol. 7, no. 3, pp. 51-57, 2003.
4. S. Zeadally, R. Wasseem, and I. Raicu, "Comparison of end-system IPv6 protocol stacks," *IEEE Proceedings Communications*, vol. 151, no. 3, 2004, pp. 238-242.
5. S. S. Mohamed, A. Y. Abusin, & D. Chieng. "Evaluation of IPv6 and Comparative Study with Different Operating Systems," *Information Technology and Applications*, vol. 2, pp. 665-670, 2005.
6. S. S. Kolahi, Z. Qu, B. K. Soorty, and N. Chand, "The Impact of Security on the Performance of IPv4 and IPv6 Using 802.11n Wireless LAN," *International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-4, 2009.
7. P. Killelea, "Web Performance Tuning," <http://www.amazon.ca/Web-Performance-Tuning-Patrick-Killelea/dp/product-description/059600172X>.
8. J. Xie, and U. Narayanan, "Performance Analysis of Mobility Support in IPv4/IPv6 Mixed Wireless Networks", *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, February 2010.

Samad Kolahi is currently a lecturer at Unitec New Zealand. Samad is a senior member of the IEEE and a committee member of the IEEE New Zealand. His research interests are local area network evaluation, IPv6, cellular networks, and VoIP. Contact him at skolahi@unitec.ac.nz.

Peng Li is a researcher at Unitec New Zealand. His research interests include wireless LAN, IPv6 networks, VoIP networks, and wireless security. Contact him at penglinz@gmail.com.