A Comparison of VoIP Performance on IPv6 and IPv4 Networks

Roman Yasinovskyy, Alexander L. Wijesinha, Ramesh K. Karne, and Gholam Khaksari
Towson University

Abstract—**We compare VoIP performance on IPv6 and IPv4 LANs in the presence of varying levels of background UDP traffic. A conventional softphone is used to make calls and a bare PC (operating system-less) softphone is used as a control to determine the impact of system overhead. The performance measures are maximum and mean delta (the time between the arrival of voice packets), maximum and mean jitter, packet loss, MOS (Mean Opinion Score), and throughput. We also determine the relative frequency distribution for delta. It is found that mean values of delta for IPv4 and IPv6 are similar although maximum values are much higher than the mean and show more variability at higher levels of background traffic. The maximum jitter for IPv6 is slightly higher than for IPv4 but mean jitter values are approximately the same. On an overloaded 100 Mbps link, packet loss can reach close to 18% for IPv4 and 24% for IPv6, and the MOS degrades significantly. At moderate levels of background traffic, the IPv4/IPv6 throughput ratio is close to the ideal (theoretical) throughput ratio, but at high levels of background traffic, throughput for IPv6 declines slightly faster than for IPv4. In general, our results indicate that the difference in VoIP performance for IPv6 and IPv4 is negligible. Results for the bare PC softphone confirm that reducing system and application overhead lowers delta and jitter values regardless of the IP version.**

Index Terms—**bare PC, call quality, IPv6, operating system, softphone, VoIP performance.**

I. Introduction

IPv6 is the next generation network layer protocol that was designed as a replacement for the current IPv4 protocol. Although IPv6 has not seen significant adoption worldwide as yet, efforts are underway in Europe and Asia to speed up migration towards IPv6, and plans for future deployment of IPv6 by U.S. government agencies also exist. While most operating systems support IPv6, uncertainty over the changes (if any) needed to run applications over IPv6, performance concerns, solutions for extending the IPv4 address space such as CIDR and NAT, and the cost of IPv6 deployment to ISPs have been factors in delaying the transition.

The primary advantages of IPv6 over IPv4 are its large address space (128-bit addresses) and potential to better support (via the traffic class field) the quality of service requirements of applications including real-time voice and video that are increasingly used on the Internet today. The increase in IPv6 packet size due to the larger addresses is partly offset by a streamlined header with optional extension headers (the header fields in IPv4 for fragmentation support and checksum are eliminated in IPv6).

As the growing popularity of VoIP will make it a significant component of traffic in the future Internet, it is of interest to compare VoIP performance over IPv6 and IPv4. The results would help to determine if there are any differences in VoIP performance over IPv6 compared to IPv4 due to overhead resulting from the larger IPv6 header (and packet size). We focus on comparing VoIP performance with IPv6 and IPv4 during the exchange of voice data only. Tests are conducted on a LAN in the presence of competing UDP traffic using a softphone running on the Linux and Windows operating systems. A softphone on a bare (operating system-less) PC is used as a control to determine the extent to which operating system, conventional protocol stack, and application overhead impacts VoIP performance over IPv6 compared to IPv4. Performance is measured using maximum and mean values of delta (the time between voice packets), maximum and mean jitter (delay variation), packet loss, MOS (Mean Opinion Score), and throughput. We also determine the relative frequency distribution of delta values and compare the IPv4/IPv6 throughput ratio with the theoretical (ideal) value.

The rest of this paper is as follows. In Section II, we discuss related work and outline the main differences between our study and previous studies of VoIP over IPv6. In Section III, we describe our test LAN and the set up for conducting experiments. In Section IV, we present and discuss the results. Section V contains the conclusion.

II. Related work

Many studies of IPv6 performance have been conducted previously. Some focus on SIP performance with IPv6, while others are concerned with IPv6-IPv4 transition issues. To the best of our knowledge, there are no studies to date that compare VoIP performance with respect to IPv6 and IPv4 focusing on voice data only using a softphone running on different operating systems.

In [1], measurements of call setup times for IPv4 and IPv6 using SIP show that native IPv4 and 6to4 tunneling perform better than native IPv6 and Teredo tunneling respectively. The authors also compare one-way delays for voice data with native IP4/IPv6 and tunneling and observe that the results are similar. Furthermore, they provide call setup times and one-way delays for proxying using SER and Mini SIP. However, their experiments only involve data passing through a single switch (i.e., there are no routers). In [2], the authors describe two IPv6-IPv4 translators for IPv6 transition: a socket layer translator for use with SIP applications on dual stacks and a SIPv6 translator designed for communication between IPv6 and IPv4 SIP user agents using SIP and RTP. Comparison of transmission delays using FTP for downloads of file sizes ranging from 10 KB to 2 GB using the socket layer translator versus native IPv6 or IPv4 shows that the overhead due to the translator is small [3]. The performance studies in this case do not involve VoIP traffic. In [4], an architecture based on SIP for integrating VoIP components in IPv4 and IPv6 networks is proposed. The authors note, based on studies using a testbed, that while the IPv6-capable SIP server (SER) and SIP IPv4-IPv6 gateway (mini-SIP-proxy) performed their functions well and the Cisco IP phone and X-Lite softphone used for IPv4 calls from/to an IPv6 user agent IPv4 were adequate, audio quality of the IPv6 softphone used (Kphone) was not satisfactory in many cases. It is not possible to compare the performance of IPv4 and IPv6 using their study since the paper did not include any quantitative results. A discussion of anticipated difficulties in deploying SIP-based services during the IP4-IPv6 transition and suggestions to mitigate them is contained in [5]. Although an overview of a proposed testbed and two experimental scenarios showing RTP and SIP flows on IPv4 and IPv6 links are given, no experimental results are included.

In [6], a performance comparison of IPv6 with respect to Windows 2003, Redhat Linux 9.0, and FreeBSD 4.9 is presented. Measurements of throughput and roundtrip time with TCP and UDP for small (32-1500 bytes) and large (8192 bytes-64 KB) file sizes show that Linux performs better than the other two operating systems. The results are limited in that the testbed consists of two workstations connected by a hub with no routers. Moreover, the conclusions regarding UDP are not directly applicable to VoIP since the packet sizes and/or rates are different from those used for voice. The performance study in [7] compares IPv6 and IPv4 implementations on Windows 2000 and Solaris 8 with respect to throughput and roundtrip time (TCP and UDP) for file sizes ranging from 64-1408 bytes, CPU utilization (TCP), socket creation time (TCP and UDP), TCP-connection time, and maximum number of TCP connections. The results show that Solaris 8 performs better than Windows 2000 and IPv6 has more overhead (or approximately equal overhead in a few cases) than IPv4. The results presented were obtained using a network consisting of two workstations directly connected by a point-to-point link. The authors also summarize the results for a network that includes two high-end commercial routers (made by IBM and Ericsson), while noting that the results are inconsistent and only serve to compare individual router performance with IPv4 and IPv6. Again, results using UDP traffic provide no insight into VoIP performance with IPv4 and IPv6.

Our study differs from (and complements) previous work in that we

- compare only VoIP performance over IPv6 and IPv4, and with respect to voice data only
- collect data on a LAN in which traffic passes through multiple switches and routers
- use a freely available softphone running on popular operating systems (Windows XP and Linux) for user machines
- provide results using a bare (operating system-less) PC softphone [8] as a control to determine the impact of reducing system, protocol stack, and application overhead.

Before presenting the results of our study, we provide a brief overview of the bare PC softphone. In a bare PC, an application runs directly over the hardware. There is no conventional operating system running in the machine and the necessary operating environment is included in the application. Thus, only essential functionality needed by the softphone application is implemented. Lean versions of the RTP, UDP, and IP protocols and an Ethernet device driver enable conventional system overhead to be significantly reduced. Details concerning the design of the bare PC softphone are given in [8]. In case of IPv6, only the elements of the protocol necessary for this study were implemented. Neighbour discovery is used to acquire the network prefix and a link local address (formed by using the Ethernet address) is assigned to the softphone. Path MTU discovery was not implemented.

III. TEST LAN AND EXPERIMENTAL SETUP

The test LAN used for experiments is shown in Figure 1. It consists of five networks, with Ethernet switches connected by Linux routers as shown. Calls are made between two softphones on clients #1 and #2 attached to opposite ends of the LAN. When calls are in progress, all voice data to client #1 is also passed via port mirroring to a Wireshark analyzer [9] connected to the end switch at client #1. Data is collected consecutively for 2 one-minute sessions. The results are obtained directly from Wireshark or computed using the data. The MOS is determined by using an online tool [10] and validated by listening.

The VoIP traffic competes with background UDP traffic sent by a pair of MGEN traffic generators [11] to a sink; these are connected to the ends of the network as shown. Both clients use 100 Mbps (Ethernet) network cards. Client #2 and the traffic generators connect to a gigabit Ethernet switch and all traffic passes through router #4 into a 100 Mbps Ethernet. This creates overloaded links and enables us to compare VoIP performance for IP4 and IPv6 under extreme conditions. All other subnets in the LAN are 100 Mbps Ethernets. We only report the results for periodic MGEN traffic streams (each stream generates 1000-byte UDP packets at 500 packets/sec), although we also conducted experiments using burst streams. Since our focus is on VoIP performance and call quality during an ongoing call, we do not compare IPv6 and IPv4 performance during call set up using SIP (this has been the subject of a previous study as discussed above).

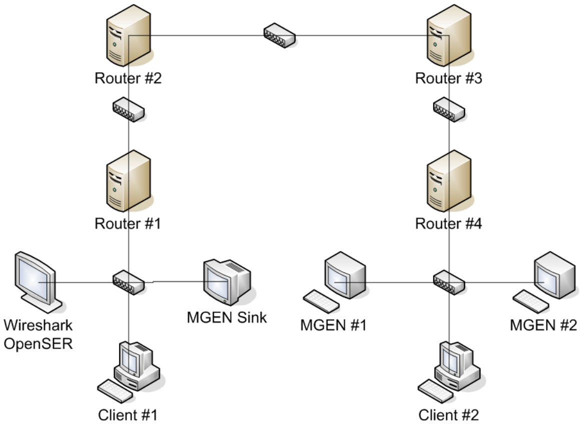

Fig. 1. Test LAN

The results reported in this paper were obtained by making calls using Linphones [12] running on Linux and Windows XP, and bare PC softphones. The reason for using Linphone was that the other open source softphones we considered (Ekiga, SIP Communicator, and Windows Messenger) either did not support IPv6 or else failed to perform consistently. All unessential system services were disabled when running experiments.

The softphones (Linphone or bare PC) as well as the Linux routers, MGEN traffic generators, and Wireshark packet analyzer ran on a Dell Optiplex GX260 PC with a 2.4 GHz Pentium 4 processor, and 512 Mb of RAM with a 3Com 10/100 external or Intel PRO /1000 internal Ethernet card (the client machines for the softphones used the external card).

The operating systems used were Fedora 7 (2.6.23.17-88), CentOS 5 (2.6.18-53.1.14), and Windows XP (SP3). The Linphone versions were 1.7.1 for Linux, and 0.5.2 for Windows XP. The MGEN 4.2b4 traffic generators ran on Windows XP, and the Wireshark 1.0 packet analyzer ran on Linux. The switches used in the test LAN were: Cisco Catalyst 2950, Netgear GS108 (1000), Netgear FS308 (10/100), and Trendnet TE100-S55E (10/100).

IV. Results and Discussion

We increased the level of MGEN background traffic from 0-200 Mbps. We did not increase beyond 200 Mbps since at this rate packet loss for voice close to 25% was seen and voice quality was poor (as confirmed by computed MOS values and listening). On the Internet, individual access links between an end user and an ISP may be a bottleneck and cause loss of VoIP quality in the presence of competing UDP traffic (assuming voice traffic is not given priority). Our results apply to situations in which total UDP traffic exceeding 100 Mbps is received on a gigabit Ethernet (such as a provider network) and needs to be forwarded along with a voice stream into a 100 Mbps Ethernet.

A. Maximum and Mean Delta

Figure 2 shows that the maximum value of delta on Linux and Windows is approximately 50 ms with no background traffic and between 100-200 ms with maximum background traffic at 200 Mbps. This implies that the maximum delay difference of consecutive packets for softphones running on the conventional operating systems is approximately 30 ms with no traffic and 80-180 ms with maximum background traffic. In contrast, the corresponding maximum delta values for the bare PC softphone are approximately 20 ms and 50 ms with background traffic i.e., the maximum delay difference varies from less than a millisecond with no background traffic to about 30 ms with background traffic. The difference between maximum delta values for IPv6 and IPv4 is 4 ms on the bare PC, 24 ms on Linux, and -31 ms on Windows.

Since delta is defined as the difference between the arrival time of consecutive packets (discounting lost packets), delta ≈ d_{i+1}-d_i+20, where d_i denotes the delay experienced by the ith packet (the packet spacing at the sender is approximately 20 ms, which is the value of delta in the ideal case when there is no delay variation). Maximum delta cannot be used as an estimate of the actual maximum delay for a packet (as seen above, delta is defined as a difference). In our experiments using the softphone on Linux, actual one-way delays varied from 0.5 to 58 ms for IPv4, and from 0.5 to 96 ms for IPv6 depending on the rate of background traffic.

Moreover, maximum delta may reflect the variation in delay only of a few packets, in which case voice quality will not be affected. Therefore, we also examined the mean value of delta over a minute of packet capture, shown in Figure 3, which confirms that mean delta values are similar for all systems and vary less than 6 milliseconds from the ideal 20 ms value. For a given system, the difference in mean delta for IPv4 and IPv6 is insignificant.

B. Relative Frequency Distribution (Delta)

The actual values of delta for IPv4 and IPv6 are better represented by the relative frequency distribution for delta. Figure 4 shows the percentage of voice packets with delta values within consecutive 25 ms intervals, and exceeding 75 ms. Several observations may be made from these results. First, it confirms that there is little difference (if any) between delta values for IPv4 and IPv6. Second, with maximum background traffic, while most packets have delta less than 50 ms, for Linux and Windows softphones some (albeit a few) packets have delta exceeding 75 ms. Even with maximum background traffic, the bare PC softphone has no packets with delta exceeding 75 ms (in fact, only a single packet for IPv6 and none for IPv4 had delta exceeding 50 ms). Third, when there is no background traffic, for both IPv6 and IPv4, only the Windows and Linux softphones had some packets with delta exceeding 25 ms (a few packets also had delta exceeding 50 ms, but none exceeded 75 ms). The voice packets sent by the bare PC softphone have little timing variation and it therefore has lower delta values for both IPv4 and IPv6. The implication is that reducing conventional operating system and application (softphone) overhead results in less intrinsic delay variation.

C. Maximum and Mean Jitter

Jitter is calculated by Wireshark according to the formula given in [13]. Maximum and minimum jitter is shown in Figures 5 and 6 respectively. Regardless of the amount of background traffic, mean jitter is essentially the same for IPv4 and IPv6 (it is identical or there is less than a millisecond difference) when the softphone is running on the same system. However, mean jitter values for the softphone running on Windows and Linux are much higher than for the bare PC softphone. While the results for maximum jitter with no background traffic are similar, maximum jitter for IPv4 and IPv6 with background traffic shows more variability even when the softphone is running on the same system (differences up to 7 ms occurred with 200 Mbps background traffic). Even with no background traffic, with the softphone running on Windows, there were always instances when jitter was 10 ms or more; this situation occurred infrequently with Linux. In contrast, jitter for the bare PC softphone with no background traffic was always less than 10 ms (jitter was almost always less than 10 ms even with background traffic; the rare instances when it was 10 ms or more occurred only under maximum load).

D. Packet Loss and MOS

Figures 7 and 8 respectively show the percentage of packet loss and the MOS as the rate of background traffic increases. At rates of up to 50 Mbps, there is no packet loss and the MOS remains at 4.41, which is to be expected since we are considering a single call and total network utilization does not exceed 50% of the capacity. When background traffic is at 100 Mbps, occasional packet loss (below 5%) is seen for all systems with IPv6 and the MOS drops to around 4. Interestingly, at the same rate of background traffic, there is no packet loss with IPv4 (except for one or two packets are lost with Windows) and the MOS is still 4.41. This shows that good voice quality may be maintained on a 100 Mbps network operating close to (or at) capacity.

When load on the 100 Mbps links exceeds capacity, packet loss increases and a greater drop in the MOS is seen. For example, when there is 150 Mbps of background traffic, packet loss for IPv4 is between 9-12%, the MOS varies from 3.1-3.5, and voice quality is good. At the maximum background traffic rate of 200 Mbps, packet loss increases and the MOS drops significantly. Specifically, packet loss for IPv4 is close to 18% for Linux and Windows, and around 12% for the bare PC. The corresponding MOS values are respectively 2.8, 2.4, and 3.2. For IPv6, packet loss and MOS are respectively 24, 17, and 20%, and 2.6, 2.4, and 2.7. Upon listening, voice quality for all systems was found to degrade significantly (and to about the same extent) with both IPv6 and IPv4 implying that better packet loss compensation techniques and codecs would be necessary when using softphones under heavily overloaded network conditions.

With both IPv4 and IPv6, packet loss was found to be more significant than delta and jitter with respect to its effect on MOS and voice quality. Thus, quality of service support in future IPv6 networks may prove to be useful in preventing VoIP packet loss and the resulting degradation in voice quality. Since the MOS is calculated based on data collected by Wireshark, it may not always accurately represent the actual voice quality perceived by the listener (the effect of jitter is eliminated or reduced by use of the jitter buffer on the softphone). In our experiments, no obvious discrepancies between computed MOS values and actual voice quality were noted.

E. Throughput

The IPv6 header size is 40 bytes and an IPv6 address is 16 bytes (twice the size of a fixed IPv4 header and address). However, if the rate of packets per second is about the same for IPv6 and IPv4, the throughput values will be similar. For example, consider the expected (theoretical) ratio of IPv4 throughput to IPv6 throughput given by

$$\frac{T_{v4}}{T_{v6}} = \frac{pps_{v4} * (d + h + 20)}{pps_{v6} * (d + h + 40)},$$

where T denotes throughput, pps denotes packets per second, d is the data size, and h is the total size of the protocol headers excluding the IP header. For VoIP, d=160 bytes assuming 20 ms of G.711 encoded voice data, and h=38 bytes (due to RTP, UDP and Ethernet including the CRC), so the expected throughput ratio ignoring all delays (so that pps is the same for IPv4 and IPv6) is 0.916 compared to a ratio of 1 for a payload of infinite size and 0.744 for no data. The expected throughput values for voice with IPv4 and IPv6 are respectively 87.2 and 95.2 Mbps (if the CRC is ignored as Wireshark does, the values are 85.6 and 93.6 Mbps respectively and the throughput ratio is 0.915). However, factors affecting throughput such as differences between IPv6 and IPv4 stacks, processing overhead, and delays are not reflected in the theoretical values.

Experimental results show that throughput declines slightly faster for IPv6 than for IPv4 under high levels of background traffic. This is because IPv6 has a lower pps. The throughput for VoIPv4 and VoIPv6 using the experimental data is compared in Figure 9. The throughput varies from 83.24-85.63 Kbps for IPv4 and 93.61-93.63 Kbps for IPv6 with no background traffic, and from 70.40-75.20 Kbps for IPv4 and 71.18-77.56 Kbps for IPv6 with maximum background traffic. The results show that for background traffic levels up to 50 Mbps, the measured throughput ratio is almost identical to the theoretical value for all systems, but varies considerably when background traffic levels are increased further. For example, the throughput ratio is almost 1 for a bare PC when the background traffic level is 200 Mbps.

V. CONCLUSION

In this paper, we compared the performance of VoIP on IPv6 and IPv4 by using the same softphone on popular operating systems. A softphone running on a bare PC was used for control purposes. Measurements of several parameters associated with call quality in the presence of varying levels of background traffic showed that there is no significant performance difference with IPv6 compared to IPv4. Measured throughput for voice data is close to the expected value only when there is moderate or no background traffic. For both IPv4 and IPv6, packet loss even under overloaded conditions results in poor voice quality and a significant drop in the MOS. VoIP performance varies considerably even for the same softphone running on different operating systems. Experiments with the bare PC softphone show that reduction of system and application overhead results in better voice quality regardless of the IP version.

VI. REFERENCES

[1] T. Hoeher, M. Petraschek, S. Tomic, and M. Hirschbichler, “Evaluating performance characteristics of SIP over IPv6,” *Journal of Networks*, vol. 2, no. 4, pp. 40-50, August 2007.

[2] W.-E. Chen, C.-Y. Su, and J.-H. Weng, “Development of IPv6-IPv4 translation mechanisms for SIP-based VoIP applications,” in *Proc. AINA’05*, vol.2, pp. 819-823, March 2005.

[3] W.-E. Chen, C.-Y. Su, and Y.-B. Lin, “NCTU SLT: A socket layer translator for IPv4-IPv6 translation,” *IEEE Communications Letters*, vol. 9(10), pp. 865-867, October 2005.

[4] L. Lambrinos and P. Kirstein, “Integrating voice over IP services in IPv4 and IPv6 networks,” in *Proc.ICCGI’07*, pp. 54-59, March 2007.

[5] M. Boucadair and Y. Noisette, “Migrating SIP-based conversational services to IPv6: complications and interworking with IPv4,” in *Proc. ICDT’07*, pp. 2-7, July 2007.

[6] S. S. Mohamed, A. Y. M. Abusin, and D. Chieng, “Evaluation of IPv6 and comparative Study with different Operating Systems,” in *Proc. ICITA’05*, vol. 2, pp. 665-670, July 2005.

[7] S. Zeadally and I. Raicu, “Evaluating IPv6 on Windows and Solaris,” *Internet Computing*, vol. 7, no. 3, pp. 51-57, May/June 2003.

[8] G. H. Khaksari, A. L. Wijesinha, R. K. Karne, L. He, and S. Girumala, “A peer-to-peer bare PC VoIP application,” in *Proc. CCNC2007*, pp. 803-807, January 2007.

[9] http://www.wireshark.org

[10] MOS Calculator, http://www.davidwall.com

[11] Networks and Communications Systems Branch, Multi-Generator(MGEN), http://cs.itd.nrl.navy.mil/work/mgen/

[12] Linphone, http://www.linphone.org

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC 3550, July 2003.

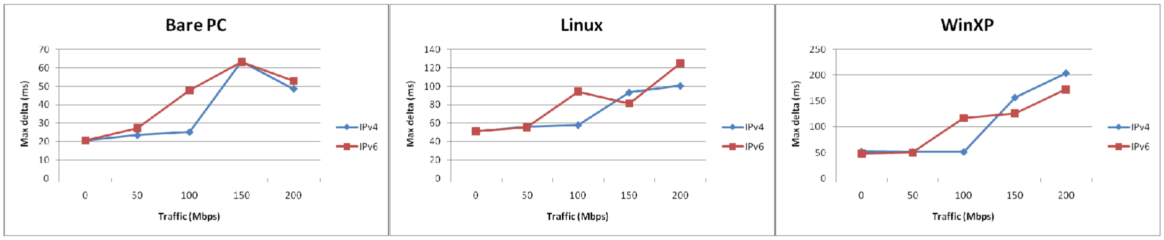

Fig. 2. Maximum delta

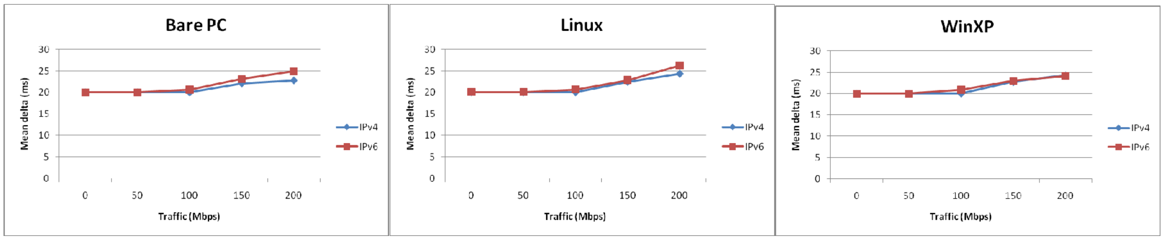

Fig. 3. Mean delta

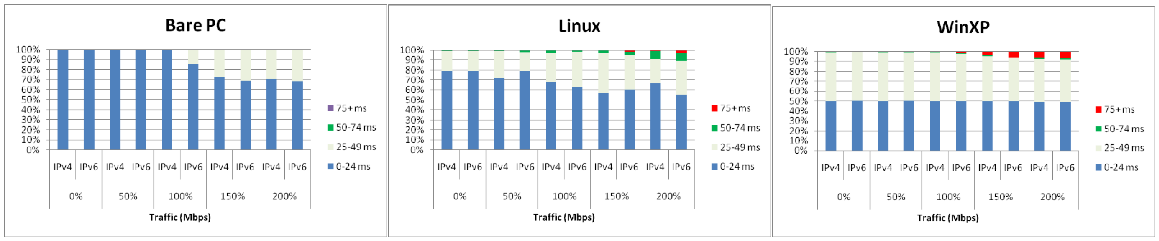

Fig. 4. Relative frequency distribution for delta

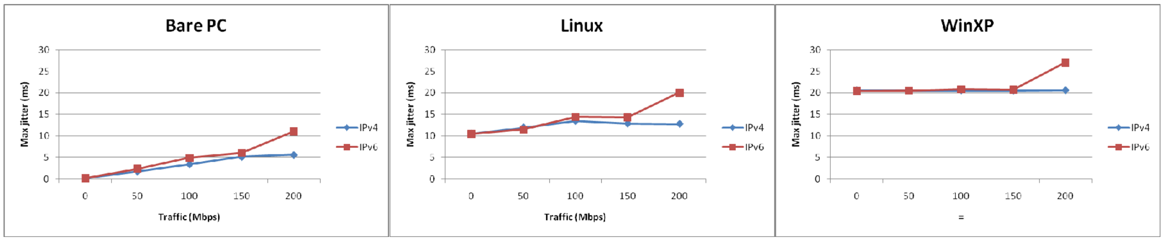

Fig. 5. Maximum jitter

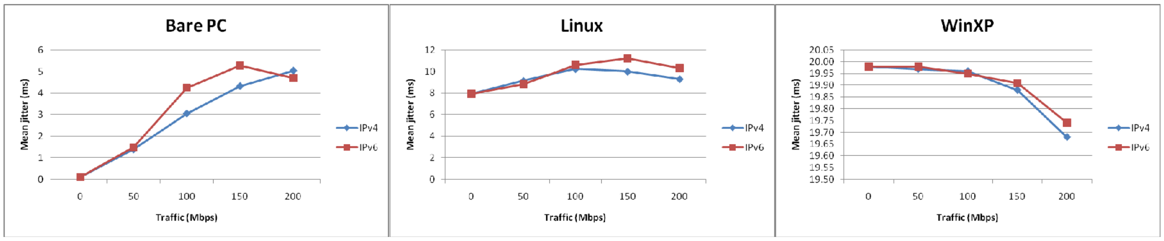

Fig. 6. Mean jitter

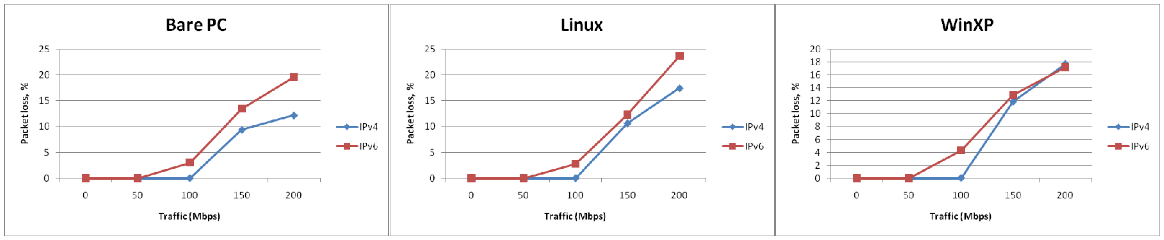

Fig. 7. Packet loss

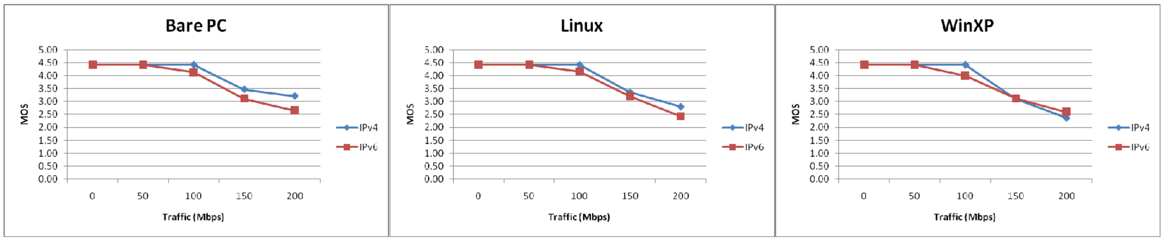

Fig. 8. MOS

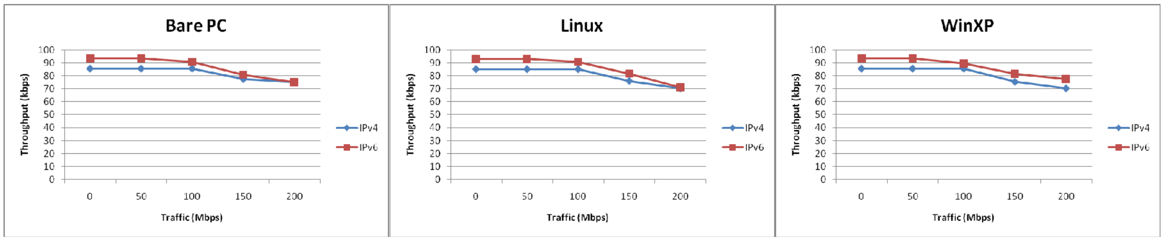

Fig. 9. Throughput