Performance of an IPv6 Web Server under Congestion

A. Loukili, A. K. Tsetse, A. L. Wijesinha, R. K. Karne, and P. Appiah-Kubi
Department of Computer & Information Sciences
Towson University
Towson, MD 21252, U.S.A.
{aloukili, awijesinha, rkarne, atsetse, appiahkubi}@towson.edu

***Abstract*-We conduct experiments using an IPv6 Web server in a test LAN environment with several routers to determine the performance under congestion due to IPv6 and IPv4 traffic. The experiments use an Apache Web server and a bare PC Web server with no operating system. Requests to the servers are made using an ordinary Web browser. Different levels of congestion are created by using MGEN traffic generators. It is found that the IPv4 throughput is slightly greater than (or approximately equal to) the IPv6 throughput under the same level of congestion. When the IPv4 throughput is larger, the differences are between 4-23%. However, Apache server delays for HTTP requests over IPv6 are between 6-32 ms more than for IPv4 depending on the level of congestion. For all congestion levels, the bare PC Web server has significantly lower throughput and larger delays than Apache regardless of whether IPv6 or IPv4 is used since it does not implement any TCP optimizations. The results show that Web server throughput and delay for browser requests depend on both the congestion traffic rate and the percentage of like traffic in the congestion mix.**

Keywords-IPv6; congestion; Web server; performance; bare PC.

I. INTRODUCTION

A recent survey of 67 top ISPs in several countries showed that 97% of them have implemented, or plan to implement the next-generation IP (IPv6) by 2013 or later [1]. Yet, due to the large number of sites including home networks that currently use IPv4 and the many IPv4 address-sensitive applications that would not work over IPv6 without code modification, IPv6 and IPv4 are likely to co-exist for an extended period of time. While IPv4 performance has been researched extensively, fewer studies deal with IPv6 performance, and with performance of servers that handle requests over both IPv6 and IPv4 when the network carries a combination of IPv6 and IPv4 traffic. We evaluate the performance of Apache and bare PC Web servers under congestion resulting from different mixes of IPv6 and IPv4 traffic by measuring network throughput and delay.

The experiments are conducted in a test LAN environment consisting of several subnets connected by routers. MGEN (MultiGenerator) traffic generators are used to create IPv4 and IPv6 background traffic at moderate and higher levels of network congestion. We use primarily TCP background traffic to reflect its predominant use in the Internet and a small amount of background UDP traffic to represent applications such as VoIP, live video, or support protocols such as DNS or DHCP. The use of a bare PC Web server with no operating system enables the impact of operating system overhead, and of not using TCP optimizations or congestion control, to be determined. The throughput and delay measurements are obtained by making requests to the Web servers using an ordinary (Firefox) Web browser. The main findings are that 1) Web server throughput is not significantly different for requests over IPv6 and IPv4; 2) Delays over IPv6 can be much larger than delays over IPv4; 3) the Apache Web server performs significantly better than the bare PC Web server for all levels of congestion; and 4) throughput and delay depend on both the congestion traffic rate and the percentage of like traffic causing the congestion.

The remainder of the paper is organized as follows. In Section II, we discuss related work. In section III, we describe the experimental set up. In section IV, we present the results. In Section V, we conclude the paper.

II. RELATED WORK

A recent study on IPv6 performance [2] concludes that RTTs for IPv6 connections are less than for IPv4, although they have higher packet loss. Higher loss over IPv6 is also noted in [3], although they find that delays over IPv6 are larger, which agrees with the studies of real-time voice and video in [4]. In [5], it is claimed that native IPv6 has significantly better throughput than IPv4 due to enhanced routing capabilities. In contrast, based on measurement studies, it is found in [6] that routing inefficiencies are the cause of poor IPv6 performance although IPv4 and IPv6 performance for data are compatible. In [7], using values of throughput, delay and other metrics in a testbed, it is determined that network performance for IPv4 and IPv6 may differ depending on traffic types and the operating system. Global Internet measurements are used to compare latency and loss over IPv4 and IPv6 in [8]. While overall performance over IPv4 is often better, about 10% of the time, latency over IPv6 can be between 10-38 ms less. Internet packet traces are used in [9] to study features of IPv6 packets. It is shown that IPv6 traffic has more self-similarity than IPv4 traffic resulting in poorer performance. In addition to the above studies comparing IPv6 and IPv4 performance, extensive studies proposing a variety of approaches for TCP congestion control [10] have also been conducted.

Our study differs from the above studies since they do not specifically determine the throughput and delay associated with browser requests over IPv6 and IPv4 to a Web server under a mix of IPv6 and IPv4 congestion traffic.

ISBN: 978-1-61208-245-5

It also differs from the study in [11] comparing the performance of Apache and IIS Web servers under congestion, which used IPv4 requests and IPv4 background traffic.

This study also uses an IPv6-IPv4 capable bare PC Web server with no operating system. Bare PC systems are based on the Dispersed Operating System Computing (DOSC) concept introduced in [12]. The bare PC C++ interfaces to the hardware used by applications (such as the bare PC Web server and client) are described in [13]. The implementation and performance of a bare PC Web server that runs over IPv4 are described in [14]. The IPv4-IPv6 capable bare PC Web server used in this study was built by modifying an IPv4 bare PC Web server.

III. EXPERIMENTAL SET UP

The network for our experiments is shown in Fig. 1. The test LAN consists of five Ethernets connected by routers. All the routers run Fedora 12 Kernel Linux 2.6.31.5-127.fc12.i686, and the network interface cards are 1 Gbps except for a 100 Mbps card on the client side of router R1. The Ethernet switches used are 1 Gbps except for switch S0, which is 100 Mbps. The 100 Mbps link and network act as a bottleneck to create congestion.

Two pairs of machines running MGEN [15] generate the background TCP and UDP congestion traffic. One machine (Dell OPTEPLEX GX260, CentOS Version 2.16.0) is a TCP source and UDP sink, and its peer (Dell OPTEPLEX GX260, Windows XP Professional 2002 SP3) is a TCP sink and UDP source to generate background IPv4 traffic. The other pair (same specifications as the first pair) serves to generate background IPv6 traffic in a similar manner. An Apache HTTP Server 2.2.16 running Fedora 12 (Constantine) Kernel Linux 2.6.31.5-127.fc12.i686 or a bare PC Web server is used as the Web server. A Firefox browser version 3.5.4 running Fedora Linux kernel 2.6.31.5-127.fc12.i686 (on the machine labeled as client in Fig. 1) makes individual requests to the Web server. The clients, servers, and routers run on Dell OPTEPLEX GX 520 PCs.

The background traffic consists of a mix of 10% UDP and 90% TCP v4 traffic, which reasonably represents the traffic composition for these protocols in the current Internet. Two different rates of background traffic are used: 75 Mbps representing moderate congestion and 100 Mbps representing higher congestion. Each rate was generated in three ways using different percentages of TCP/UDP traffic over IPv4 and IPv6 while maintaining the overall 90/10% TCP/UDP mix. These percentages with their respective compositions of IPv4 and IPv6 traffic are shown in Table I, where the percentages for 75 Mbps are labeled as congestion levels C1B, C2B, and C3B, and those for 100 Mbps are labeled as C4B, C5B, and C6B.

Note that percentages of IPv4 and IPv6 TCP/UDP traffic for levels C1B and C4B are equal. Likewise, the percentage of IPv6 TCP/UDP traffic for levels C2B and C5B is three times that of IPv4 traffic, and the reverse is true for levels C3B and C6B. The amounts of TCP data carried in the MGEN packets are 1440 and 1460 bytes respectively for IPv6 and IPv4, and MGEN is configured to use 10 flows at different rates to achieve the desired levels of congestion.

The throughput and delay when the Firefox Web browser requests the 320 KB file from the Apache or bare PC Web server were determined by using the network protocol analyzer Wireshark [16] (with port mirroring) at the server side (i.e., connected to switch S4). The throughput ignores retransmissions, but considers both incoming and outgoing traffic associated with a single request. From the Wireshark traces, the values of the delay were computed by using the time stamps for the HTTP Get request and the last valid ack sent by the client (before the FIN+ACK). While the throughput and delay are related, the delay is the delay for the data transfer only i.e., it is not the total delay for the request since it is not measured from the TCP SYN. Each experiment was repeated ten times (for each of IPv4 and IPv6), and the average values of the delay were used.

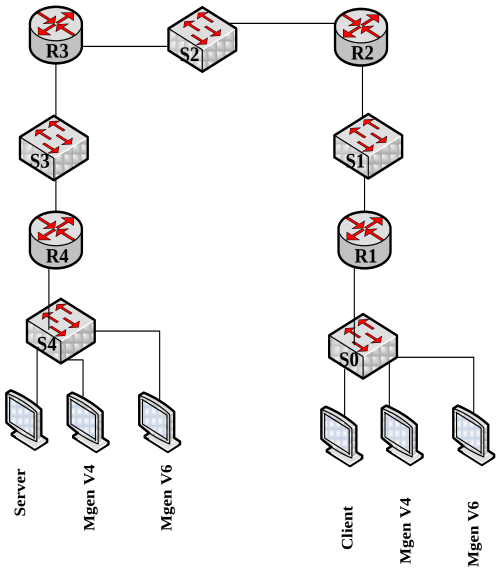

Figure 1. Test network

TABLE I. BACKGROUND TRAFFIC CONDITIONS

Congestion Level	Percentages of V4 and V6 TCP/UDP Traffic				Rate (Mbit s/sec)
	TCP V4	TCP V6	UDP V4	UDP V6	
C1B	45	45	5	5	75
C2B	22.5	67.5	2.5	7.5	75
C3B	67.5	22.5	7.5	2.5	75
C4B	45	45	5	5	100
C5B	22.5	67.5	2.5	7.5	100
C6B	67.5	22.5	7.5	2.5	100

IV. RESULTS

A. Apache Throughput

The throughput associated with a single browser request to the Apache server was obtained as described above. The results are shown in Fig. 2. It is seen that the throughput for IPv4 is slightly higher than for IPv6 except for congestion level C3B when they are approximately equal (IPv6 throughput is higher, but the difference is only about 0.4 Mbps). The other differences range from approximately 1 Mbps (for congestion level C6B) to 7 Mbps (for congestion level C2B).

It can also be seen from the figure that throughput values range from about 20-27 Mbps for IPv6 and from about 22-31 Mbps for IPv4. The highest throughput is with congestion level C3B and C2B for IPv6 and IPv4 respectively, and the lowest throughput is with level C4B for both IPv6 and IPv4. So the IPv6 throughput is highest when the congestion traffic is lower (75 Mbps) and its percentage of IPv6 traffic is lower (25%), and it is lowest when the congestion traffic is higher (100 Mbps) and its percentage of IPv6 traffic is equitable (50%). Similarly, the IPv4 throughput is highest when the congestion traffic is lower and its percentage of IPv4 traffic is lower, and it is lowest when the congestion traffic is higher and its percentage of IPv4 traffic is equitable. For both IPv6 and IPv4, the throughput is highest when congestion traffic is lower and like traffic is lower, and it is lowest when congestion traffic is higher and like traffic is 50% or more. Also, the low percentage of UDP traffic appears to have a negligible impact on throughput as would be expected.

B. Apache Delay

In the absence of congestion traffic, the delay for the file transfer from Apache to the browser was found to be 48 milliseconds for both IPv4 and IPv6. Fig. 3 shows the transfer delay when the browser makes the request under each of the six congestion levels C1B-C6B. The delays during congestion are much larger, between 84-103 ms for IPv4 and between 105-121 ms for IPv6. Also, the delay difference between the IPv4 and IPv6 delays (for a given congestion level) varies from 6-32 ms. The difference between IPv6 and IPv4 delays are largest for congestion levels C2B and C5B, which have larger percentages of IPv6 traffic (75%), and smallest for level C3B, which has lower congestion traffic (75 Mbps) and a lower percentage of IPv6 traffic (25%).

The highest delay is with congestion level C4B and C6B for IPv6 and IPv4 respectively, and the lowest delay is with level C3B and C2B for IPv6 and IPv4 respectively. It can be seen that the IPv6 delay is highest when the congestion traffic is higher (100 Mbps) and its percentage of IPv6 traffic is equitable (50%), and it is lowest when the congestion traffic is lower (75 Mbps) and its percentage of IPv6 traffic is lower (25%). Similarly, the IPv4 delay is highest when the congestion traffic is higher and its percentage of IPv4 traffic is higher, and it is lowest when the congestion traffic is lower and its percentage of IPv4 traffic is lower. For both IPv6 and IPv4, the delay is highest when congestion traffic is higher and like traffic is 50% or more, and it is lowest when congestion traffic is lower and like traffic is lower.

The above results using the Apache server and an ordinary browser suggest that the throughput and delay for a single request over IPv6 or IPv4 in a network with mixed congestion traffic are affected by two factors. First, the rate of congestion traffic, and second, by the percentage of like traffic in the congestion mix. It should also be noted that these results are for the case when the server only receives a single request from a browser i.e., there are no additional delays at the server due to any other requests.

C. Bare PC Web Server Throughput

We now consider bare PC Web server performance. Fig. 4 shows the throughput for a browser request from Firefox under the six congestion levels. It can be seen that throughput is much lower than for the Apache server under the same congestion levels. The throughput for IPv4 is slightly higher than for IPv6 as seen above for the Apache server. It ranges from 3.4-4.8 Mbps for IPv6 and from 4.2-5.3 Mbps for IPv4. The difference between throughput for different congestion levels with IPv6 or IPv4 is small. For example, the throughput difference for levels C2B and C3B with IPv6, and for levels C1B and C2B with IPv4, is only 0.1 Mbps. The difference between IPv6 throughput and IPv4 throughput ranges from 0.3 Mbps (for congestion level C3B) to 1.4 Mbps (for congestion level C5B). It can be seen that the differences between IPv4 and IPv6 throughput under a given level of congestion are much smaller than for Apache. The IPv6 throughput is highest for congestion level C1B and lowest for level C5B, while the IPv4 throughput is highest for congestion level C2B and lowest for level C6B.

The likely reason that the bare PC Web server has lower throughput than Apache under congestion is the absence of TCP optimizations. In particular, the bare PC server does not use selective acks (i.e. TCP SACK) or fast retransmit/recovery, and retransmits all the data after waiting for a timeout in the event of a loss. However, the bare PC Web server implementation does not have separate IPv4 and IPv6 stacks unlike Linux and other conventional servers. Instead, it uses a single RCV (Receive) task to process an incoming IP packet regardless of whether it is an IPv6 packet or an IPv4 packet. Also, the HTTP and TCP code in the server is intertwined. This implies that there is little difference in overhead when processing the two types of IP packets. However, more studies with a bare PC server that implements TCP optimizations is needed to determine the extent of possible throughput improvement for IPv6 and IPv4 due to eliminating operating system overhead. Similarly, since the performance of the bare PC server is worse than that of Apache, the impact of not implementing any congestion control mechanisms is not known.

The IPv6 throughput is highest when the congestion traffic is lower (75 Mbps) and its percentage of IPv6 traffic is equitable (50%), and it is lowest when the congestion traffic is higher (100 Mbps) and its percentage of IPv6 traffic is higher (75%). Similarly, the IPv4 throughput is highest when the congestion traffic is lower and its percentage of IPv4 traffic is lower, and it is lowest when the congestion traffic is higher and its percentage of IPv4 traffic is higher. For both IPv6 and IPv4, the throughput is highest when congestion traffic is lower and like traffic is 50% or less, and it is lowest when congestion traffic is higher and like traffic is higher. This is similar to the results seen for Apache.

D. *Bare PC Web Server Delay*

The delays to transfer the 320 KB file to the browser from the bare PC server in the absence of congestion traffic are 99 milliseconds and 116 milliseconds over IPv4 and IPv6 respectively (more than double the delays for Apache). When there is congestion traffic, the delays (shown in Fig. 5) range from 727-824 ms for IPv6 and from 654-725 ms for IPv4, which are much larger than the corresponding delays for Apache. Also, it can be seen that IPv6 delay are larger than the IPv4 delays as for Apache, but the delay differences between IPv6 and IPv4 for all congestion levels are now much higher than for Apache. These differences range from 64-118 ms. As with the lowered throughput, the increased delays are likely due to the absence of any TCP optimizations.

The difference between IPv6 and IPv4 delays is largest for congestion level C5B, which has a larger percentage of IPv6 traffic (75%), and smallest for level C3B, which has lower congestion traffic (75 Mbps) and a lower percentage of IPv6 traffic (25%). The lowest delay is with congestion level C3B for IPv6 and with level C2B for IPv4, and the highest delay is with congestion level C5B for IPv6 and level C6B for IPv4. It can be seen that the IPv6 delay is highest when the congestion traffic is higher (100 Mbps) and its percentage of IPv6 traffic is higher (75%), and it is lowest when the congestion traffic is lower (75 Mbps) and its percentage of IPv6 traffic is lower (25%). Similarly, the IPv4 delay is highest when the congestion traffic is higher and its percentage of IPv4 traffic is higher, and it is lowest when the congestion traffic is lower and its percentage of IPv4 traffic is lower. For both IPv6 and IPv4, the delay is highest when congestion traffic is higher and like traffic is higher, and it is lowest when congestion traffic is lower and like traffic is lower.

These results for bare PC server throughput and delay are similar to the results for the Apache server, although the throughput is much lower and the delays are much higher. As with Apache, the throughput and delay depend on both the congestion rate of congestion traffic and the percentage of like traffic causing the congestion.

V. CONCLUSION AND FUTURE WORK

We studied the performance of IPv6 Web servers under different levels of congestion. Studies were conducted in a test LAN with several routers, and used a conventional Apache Web server and a bare PC Web server with no operating system. HTTP requests over IPv6 and IPv4 were sent to the servers using an ordinary Web browser. The results for both servers show that throughput over IPv6 throughput is slightly lower than or approximately equal to IPv4 throughput depending on the congestion level, whereas delays are much higher over IPv6 than over IPv4. For all congestion levels, bare PC server throughput and delay are significantly worse than for Apache due to not implementing any TCP optimizations. Studies using bare PC Web servers with the usual TCP optimizations and congestion control mechanisms will enable the overhead due to an operating system to be determined. The results of this study show that the performance of an IPv6 Web server for requests over IPv6 and IPv4 depend on both the congestion traffic rate and the percentage of like IP traffic.

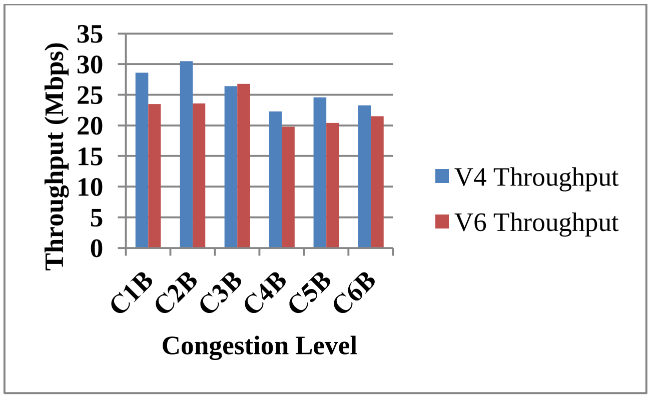

Figure 2. Apache throughput under congestion

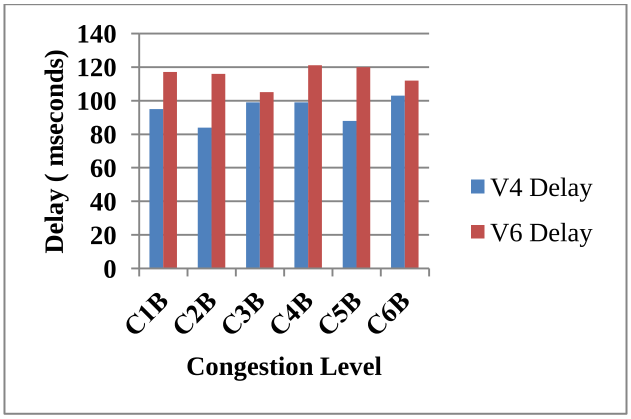

Figure 3. Apache delay under congestion

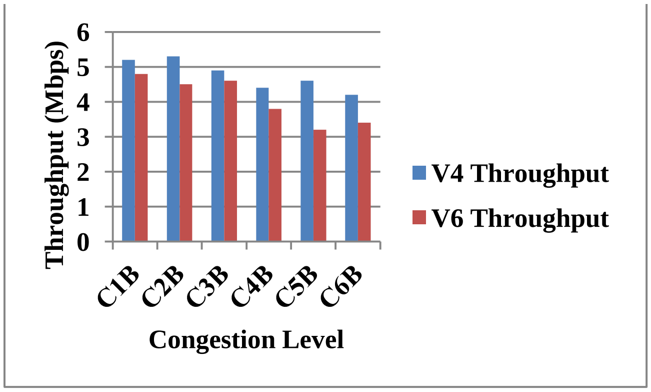

Figure 4. Bare server throughput under congestion

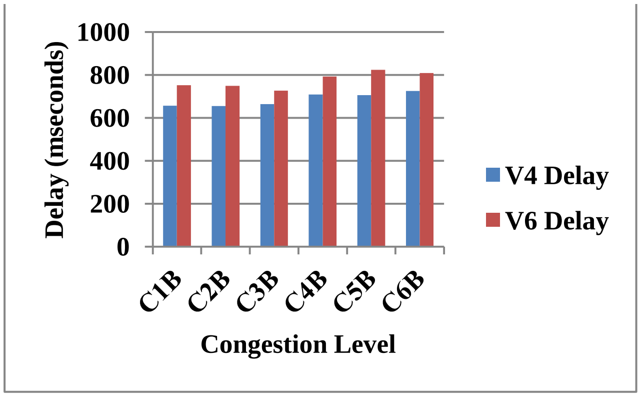

Figure 5. Bare server delay under congestion

References

[1] Nominum Survey, http://www.nominum..com/, accessed: November 24, 2012.

[2] Y. Wang, S. Ye, and X. Li, "Understanding Current IPv6 Performance: A Measurement Study," 10th IEEE Symposium on Computers and Communications (ISCC 2005), pp. 71-76, 2005.

[3] X. Zhou, M. Jacobsson, H. Uijterwaal, and P. Van Mieghem, "IPv6 delay and loss performance evolution," International Journal of Communication Systems, vol. 21, no. 6, pp. 643–663, 2008.

[4] Md. T. Aziz and M. S. Islam, "Performance Evaluation of Real–Time Applications over DiffServ/MPLS in IPv4/IPv6 Networks," Masters Thesis, Blekinge Institute of Technology, 2011.

[5] D. T. Ustundag, "Comparative routing performance analysis of IPv4 and IPv6," Masters Thesis, Atilim University, 2009. http://acikarsiv.atilim.edu.tr/browse/100/301.pdf, accessed: July 22, 2012.

[6] M. Nikkhah, R. Guerin, Y. Lee, and R. Woundy, "Assessing IPv6 through web access a measurement study and its findings," 7th Conference on Emerging Networking Experiments and Technologies (CoNEXT '11), pp. 1-12, 2011.

[7] S. Narayan, P. Shang, and N. Fan, "Performance Evaluation of IPv4 and IPv6 on Windows Vista and Linux Ubuntu," International Conference on Networks Security, Wireless Communications and Trusted Computing, (NSWCTC '09), pp. 653-656, 2009.

[8] A. Berger, "Comparison of performance over IPv6 versus IPv4," Intenet Statistics and Metrics Anaysis (ISMA 2012) AIMS-4 Workshop on Active Internet Measurements, pp. 1-30, 2012. http://www.caida.org/workshops/isma/1202/slides/aims1202_acox_supplement.pdf, accessed: July 22, 2012.

[9] C. Ciflikli, A. Gezer, and A. T. Ozsahin, "Packet traffic features of IPv6 and IPv4 protocol traffic," Turk. J. Elec. Eng. & Comp. Sci., vol. 20, no. 5, pp. 727-749, 2012.

[10] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-Host Congestion Control for TCP," IEEE Communications Surveys and Tutorials, vol. 12, no. 3, pp. 304-342, 2010.

[11] A. Loukili, A. L. Wijesinha, R. K. Karne, and A. K. Tsetse, "Web Server Performance with Cubic and Compound TCP," 7th International Conference on Communication, Internet, and Information Technology (CIIT), 2012.

[12] R. K. Karne, K.V. Jaganathan, T. Ahmed, and N. Rosa, "DOSC: Dispersed Operating System Computing," 20th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '05 Onward Track), pp. 55-61, 2005.

[13] R. K. Karne, K.V. Jaganathan, and T. Ahmed. "How to run C++ applications on a bare PC," 6th ACIS International Conference on Software Engineering, Networking, Parallel/Distributed Computing (SNPD 2005), pp. 50-55, 2005.

[14] L. He, R. K. Karne, and A. L. Wijesinha, "Design and performance of a bare PC Web Server," International Journal of Computer Applications, pp. 100-112, vol. 15, no. 2, June 2008.

[15] Multi-Generator (MGEN), http://cs.itd.nrl.navy.mil/work/mgen, accessed: July 22, 2012.

[16] Wireshark packet analyzer, http://www.wireshark.org/, accessed: July 22, 2012.