Impact of IPv6 on End-User Applications

*Ioan Raicu
Department of Computer Science
Purdue University
1398 Computer Science Building
West Lafayette, IN 47907, USA
iraicu@cs.purdue.edu

Sherali Zeadally
Department of Computer Science
Wayne State University
5143 Cass Ave., 454 State Hall
Detroit, MI 48202, USA
zeadally@cs.wayne.edu

ABSTRACT

With IPv6's maturity increasing, there is a need to evaluate the performance benefits or drawbacks for end-users that the new IPv6 protocol will have compared to the well-established IPv4 protocol. Theoretically, the expected overhead between the two different protocols should be directly proportional to the difference in the packet's header size, and therefore the expected performance of IPv6 should be similar to IPv4. However, according to our findings, the empirical performance difference between IPv4 and IPv6 in a real setting is much higher than anticipated. Our experimental setup is the key ingredient to our findings since it demonstrates the current performance of IPv6 as delivered by commercial routers supporting IPv6 for performance metrics such as throughput and latency. We hope that our experience and results will be useful to end users who are planning migration to IPv6 as well as designers and implementers of IPv6 (router and host implementations).

Keywords: IPv4, IPv6, Operating System, Networking, Protocol, Performance.

1 INTRODUCTION

It is a well-known fact that today's networks, mainly the Internet has surpassed IPv4 (Internet Protocol Version 4) [1, 2] capabilities. The shortcomings of IPv4 were seen well in advance, and therefore work started almost a decade ago. Its successor will be IPv6 (Internet Protocol Version 6) [3, 4, 2], and according to most experts, over the next five to ten years, IPv6 will be slowly integrated into the existing IPv4 infrastructure. [2]

IPv6 hopes to solve some of IPv4's shortcomings. Our work focuses on evaluating the performance of the IPv6 protocol using traditional data transfers. Many of IPv6's features supporting QoS (Quality of Service) traffic are not used in our experiments, but will be investigated in future works.

The remainder of this paper is structured as follows. Section 2 covers some background information about the fundamental differences between IPv4 and IPv6. In section 3, we discuss related work. Section 4 describes the testbed configurations, experimental procedures, and performance metrics used. In section 5, we discuss our experimental results. Finally, in section 6, we present our conclusions.

2 BACKGROUND

Internet Protocol was first developed in the early 1980s. In the early 1990s, it became pretty evident that if the Internet will continue to grow at the rate it was growing, the IPv4 address space would be depleted by the turn of the millennium. Some temporary solutions were offered, such as NAT (Network Address Translator) [5] or CIDR (Classless InterDomain Routing) [2], however work began on a new Internet Protocol, which was first called IPnG from Internet Protocol Next Generation, but later became known as IPv6, Internet Protocol version 6.

The main reason for the deployment of a new version of the Internet Protocol was to increase the address space. IPv6 was designed to use a 128-bit address scheme rather than the 32-bit address used in IPv4[6]. There were other reasons driving the deployment of IPv6 just as hard as the address space depletion problem. Twenty years ago, the only kind of traffic that existed on the Internet was elastic traffic, such as emails or file transfers. This kind of traffic is very flexible regardless of the network conditions; on the other hand, inelastic traffic requires a certain level of performance, which if it cannot be met, the data stream is rendered useless. In the past decade, multimedia applications have emerged and have mostly dominated the Internet's growth and demand for more bandwidth and processing power. IPv6 was designed to efficiently support both elastic and inelastic traffic and also address issues such as scalability, security, and support for multimedia transmissions. Overall, IPv6 was carefully thought out and was designed with future applications in mind. [2]

Theoretically, a close look at the breakdown of the various headers in both IPv4 and IPv6, we note that the

---

* *The work presented herein was completed by the author while he was a graduate student in the Department of Computer Science at Wayne State University, Detroit, Michigan, USA.*

0-7803-7661-7/03/$17.00©2003 IEEE

overhead difference incurred between IPv4 and IPv6 is minimal. From Table 1, the primary difference between IPv4 and IPv6 is that IPv4 has a 20 byte header while IPv6 has a 40 byte header. Although the address space in IPv6 is four times the size of its counterpart, IPv6 has decreased the number of required fields and made them optional as extension headers. Let's take the IPv4 UDP packet as an example to better understand Table 1. The total Ethernet MTU is 1514 bytes, from which 14 bytes are the Ethernet header, 20 bytes are the IP header, and 8 bytes are the UDP header. The payload for a UDP packet in IPv4 is 1472 bytes, and is computed by: MTU = Payload + TLH + NLH + DLLH. The payload is the application layer data size; TLH is the transport layer (TCP/UDP) header size; NLH is the network layer (IP) header size; DLLH is the data link (Ethernet) layer header size; MTU is the total Ethernet MTU size that is transmitted on the physical medium.

The overhead incurred due to the header information can be calculated by dividing the TCP or UDP payload size by the Ethernet MTU size. For example, the difference between IPv4 UDP and IPv6 UDP is a mere 1.42 %, while for TCP it is 1.44 %.

	IPv4 TCP	**IPv6 TCP**	**IPv4 UDP**	**IPv6 UDP**
TCP/UDP Payload	1460	1440	1472	1452
TCP/UDP Header	20	20	8	8
IP Payload	1480	1460	1480	1460
IP Header	20	40	20	40
Ethernet Header	14	14	14	14
Total Ethernet MTU	1514	1514	1514	1514
Overhead %	3.7%	5.14%	2.85%	4.27%

Table 1: The overhead incurred by header information.

In theory, the performance overhead between these two protocols appear to be minimal, however as we will discuss in Section 5, the real performance difference between IPv4 and IPv6 proved to be quite larger than the predicted theoretical difference.

3 RELATED WORK

Our work was driven by the fact that there were no performance comparisons between IPv4 and IPv6 in a real world setting using routers with IPv6 support. Even if some of the experiments in the research community used routers, they were always software-based routers built from conventional PCs to handle the necessary routing tasks. Most of the industry wide routers implement most of their functionality in hardware and therefore are much more efficient than a software router implementation. The reason few researchers tested IPv6's performance using real routers is because hardware-based routers supporting dual stack IPv4/IPv6 are expensive; as an example, the two routers we used for our experiments cost a total of US $60,000 at the time we conducted these experiments.

Another contribution of this work different from previous works is that we compare two different implementations of IPv6 running on Solaris 8 and Windows 2000 operating systems. Our performance metrics included throughput and latency. Note that these metrics influence the perceived performance of the network the most.

The following paragraphs briefly describe some of the related works that had similar goals to our own. In [7], the first attempt at developing an IPv6 protocol stack for Windows NT is described. The work presented only offered a performance evaluation of a small subset of tests (only throughput) that we performed. They also had no router and hence only connected the two PCs back-to-back using a point-to-point link.

In [8], the author evaluated the Microsoft Research (MSR) IPv6 BETA protocol stack for Windows NT 4.0. The performance of was measured by analyzing its network latency, throughput, and processing overheads. Their testbed consisted of two Pentium machines with 100Mbps fast Ethernet connected via an unloaded switch. This work only evaluated IPv6 and did not compare it with IPv4. Furthermore, they only evaluated the Windows NT implementation and did not compare it with any other IPv6 implementations. There were no routers used in their experiments either.

In [9], the authors evaluate the performance of data transmission over IPv4 and IPv6 using various security protocols. They utilized end hosts with FreeBSD 2.2.8 and KAME IPv6 protocol stack and a router implemented in a PC platform also running FreeBSD 2.2.8 and KAME IPv6 protocol stack.

In both [10, 11], the authors presented an evaluation of IPv6 compared to IPv4 using the dual stack implementation of KAME over FreeBSD OS using the ping utility and a FTP application; their metrics were latency and file transfer throughput. They used a ported FTP application to find out the throughput rates of the IPv6 protocol; they used the ping utility to find the latency. In [11], they had no router, but rather connected the two end hosts via a hub. In [10], they utilized a software-based router running FreeBSD. They did not experiment with parameters, such as buffer size, packet size, and of course they could not perform any UDP tests due to the nature of FTP.

4 MEASUREMENT PROCEDURES AND TESTBED

4.1 Testbed Configuration

Our testbed consisted of two dual stack (IPv4/IPv6) routers: an Ericsson AXI 462, and an IBM 2216 Nways Multiaccess Connector Model 400. Dual stack implementation specifications can be found in Request for Comments 1933 [12]. We had two identical workstations that were connected directly to the routers and were configured to be on separate networks. Each router supported two separate networks each.

Both workstations were equipped with Intel Pentium III 500 MHz processors, 256 megabytes of SDRAM PC100, two 30GB IBM 7200 RPM IDE hard drive, and COM 10/100 PCI network adapters. The workstations were loaded with both Windows 2000 Professional and Solaris 8.0 as a dual boot configuration on two separate and identical hard drives. Windows 2000 had the IPv4 stack as a standard protocol; however in order to get IPv6 support, an add-on package was installed. There were two choices, both written by Microsoft and they were both in Beta testing. We chose the newer release of the two, “Microsoft IPv6 Technology Preview for Windows 2000” [13] which is supported by Winsock 2 as its programming API. It was evident that Microsoft’s IPv6 stack for Windows 2000 is not in production yet since it had various deficiencies. It did not seem to handle fragmentation well for the UDP transport protocol, and therefore we limited our test to message sizes less than the Ethernet MTU size of 1514 bytes. It also does not support IPSec yet, but that was outside of the scope of this paper and therefore IPSec was not as relevant for our work. On the other hand, Solaris 8.0 came with a dual production level IPv4/IPv6 stack. Because of Microsoft’s IPv6 limitation with fragmentation, the tests on Solaris were limited to 1514 byte UDP messages as well.

To better understand the relevance of our results, we developed three testbed configurations:

- IBM-Ericsson Testbed depicted in Figure 1: the end hosts are connected to each other via both the IBM router and the Ericsson router
- Ericsson Testbed depicted in Figure 2: the end hosts are connected to each other via the Ericsson router
- IBM Testbed depicted in Figure 3: the end hosts are connected to each other via the IBM router

The first configuration in Figure 1 depicts the entire testbed utilizing two routers in between the two end hosts. This scenario is the most realistic and most likely to be found in a real world setting of our three testbeds. Note that the IP addresses of the end hosts are not on the same network anymore, and hence we have the routers to allow communication between the two separate networks. On the Ericsson router, R3 through R6 are the various network cards available (we only used cards R3 and R4 for our experiments); each interface card has both an IPv4 and an IPv6 address. Similarly, on the IBM router, R1 through R8 are the various network cards that are available (we only used cards R4 and R8 for our experiments); each interface card has both an IPv4 and an IPv6 address.

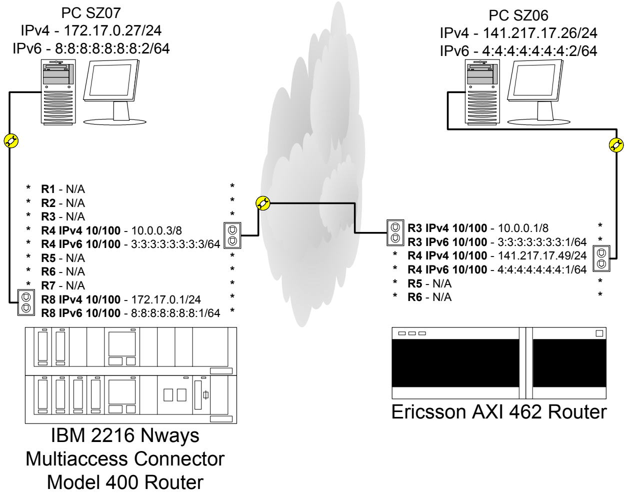

Figure 1: IBM-Ericsson Testbed architecture; two routers are depicted, an IBM 2216 Nways Multiaccess Connector Model 400 and an Ericsson AXI 462

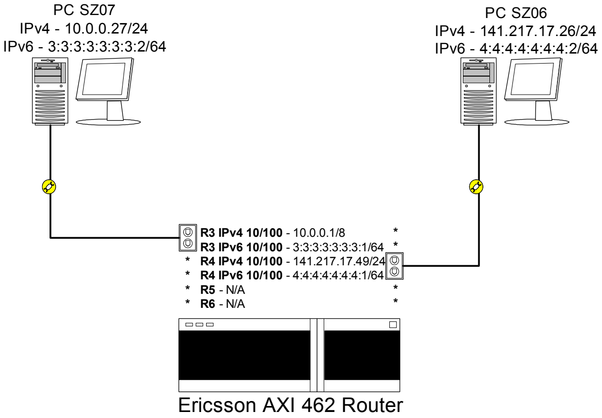

Figure 2: Ericsson Testbed architecture; one router configuration is depicted using the Ericsson AXI 462

For completeness and a thorough understanding of the results, we designed two more testbeds, which are comprised of two workstations and one router. This is depicted in Figure 2, which has two end PCs (SZ06 and SZ07) that are directly connected to the Ericsson router. Our last testbed is depicted in Figure 3 as we left out the Ericsson router and replaced it with the IBM router to connect the workstations together.

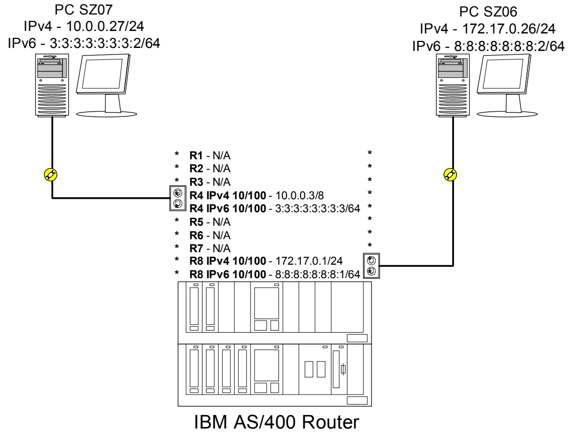

Figure 3: IBM Testbed architecture; one router configuration is depicted using the IBM 2216 Nways Multiaccess Connector Model 400

A fourth point-to-point configuration was also used, in which no routers were used and experiments were conducted between the two hosts directly over a twisted Ethernet cable. Due to the length constraints of this paper, these results will not be presented here, but can be found in [14].

4.2 Measurement Procedures

Our metrics of evaluation were throughput and latency. All the performance measurement software was written in C++.

The majority of the tests were done for a period of about 60 seconds, which netted about 50,000 packets to about 1,000,000 packets, depending on the size of the packets sent and what tests were being completed. The tests dealing with testing the throughput of the UDP transport protocol were limited to 1472 byte datagrams because of a potential undocumented fragmentation problem in the IPv6 protocol stack. All other tests were done using various packet sizes ranging from 64 bytes to 64 Kbytes. Each test was repeated three times in order to avoid any inconsistencies. On occasions when the different tests were not consistent enough to have a solid conclusion, the experiments were performed several more times until there was enough data to conclude our findings.

- **Throughput:**
  The rate at which bulk data transfers can be transmitted from one host to another over a sufficiently long period of time (Mbit/s).
- **Latency**
  Latency, also known as RTT (round trip time), is the amount of time it takes one packet to travel from one host to another and back to the originating host (RTT in microseconds)

5 EXPERIMENTAL RESULTS

5.1 IBM-Ericsson Testbed Performance Results

5.1.1 Throughput

As Figure 4 and Figure 5 indicate, it can be seen that Solaris 8.0 performs slightly better than Windows 2000 over the entire packet size range. However, when comparing IPv4 and IPv6, IPv6 outperforms IPv4 by as much as 300%.

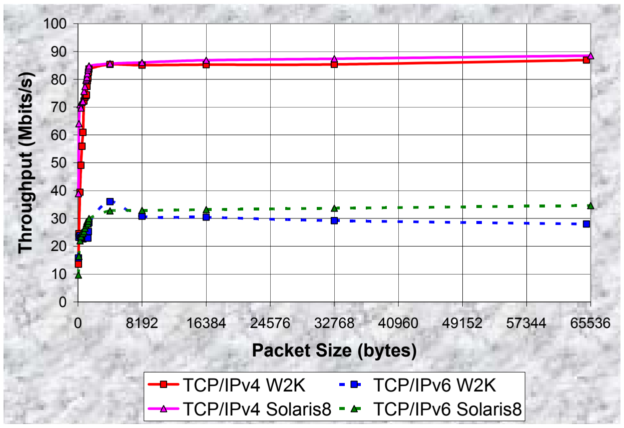

Figure 4: IBM-Ericsson Testbed: TCP throughput results with packet size ranging from 64 bytes to 64 Kbytes

As a quick overview, the dotted lines represent the IPv6 protocol while the solid lines represent the IPv4 protocol. It should be evident that if IPv4 achieves throughput rates surpassing 88 Mbit/s while IPv6 barely gets over 34 Mbit/s under Solaris and 28 Mbit/s under Windows, the performance overhead incurred will render IPv6 as unappealing. Under Windows, TCP's overhead surpasses 250% for the throughput experiment. For Solaris, the overhead is as high as 300% for very small packet sizes and the best it can do is about 150% overhead for larger packet sizes.

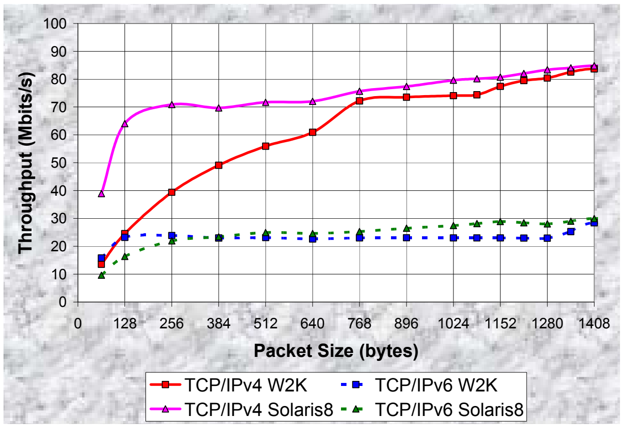

Figure 5: IBM-Ericsson Testbed: TCP throughput results with packet size ranging from 64 bytes to 1408 bytes

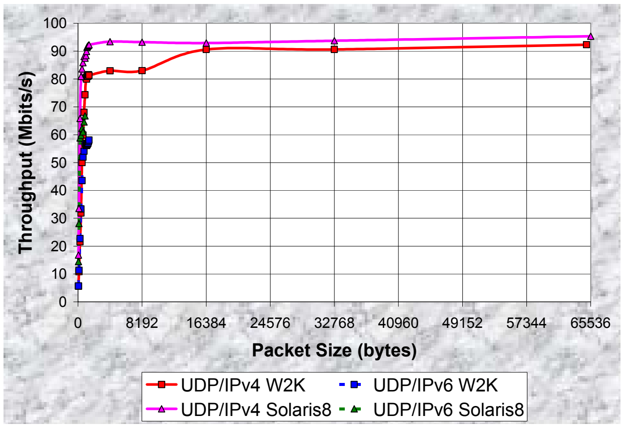

Figure 6: IBM-Ericsson Testbed: UDP throughput results with packet size ranging from 64 bytes to 64 Kbytes

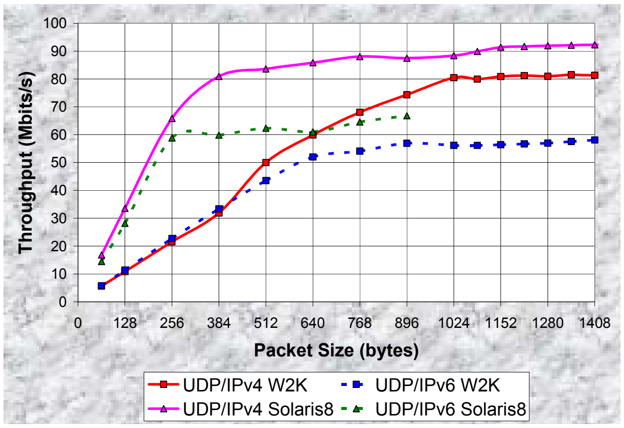

Figure 7: IBM-Ericsson Testbed: UDP throughput results with packet size ranging from 64 bytes to 1408 bytes

As Figure 5 indicates, it is evident that starting from small packet sizes, the IPv6 protocol performs very poorly under both Windows and Solaris. From [14], we know that IPv6 incurs minimal overhead without any routers; therefore, the routers are the main cause of the poor performance of IPv6 in this experiment.

In order to examine this further, we tried to measure each router's individual performance by repeating the same experiments with only one router instead of both. Those findings will be presented in section 5.2 and 5.3.

5.1.2 Latency

Figure 8 clearly depicts similar performance deficits for IPv6 in the larger packet sizes. Notice how both Windows and Solaris offer nearly identical performance.

However, for 64 Kbyte packets, IPv6 has a latency of about 40,000 microseconds (40 milliseconds) while IPv4 retains the fairly low 14,000 microseconds. In evaluating the latency performance, it is beginning to make sense why the throughput performance of IPv6 was so bad under the TCP transport protocol.

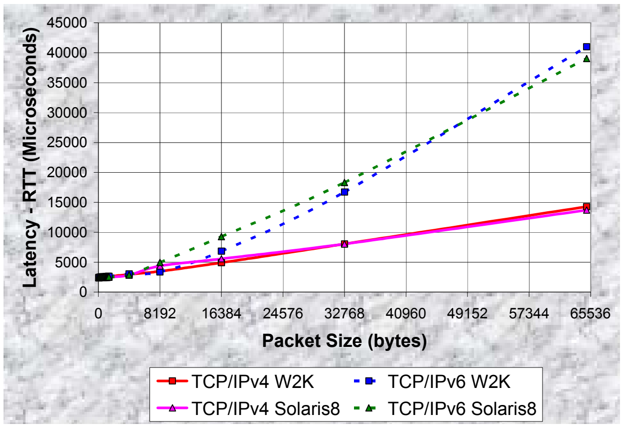

Figure 8: IBM-Ericsson Testbed: TCP latency results with packet size ranging from 64 bytes to 64 Kbytes

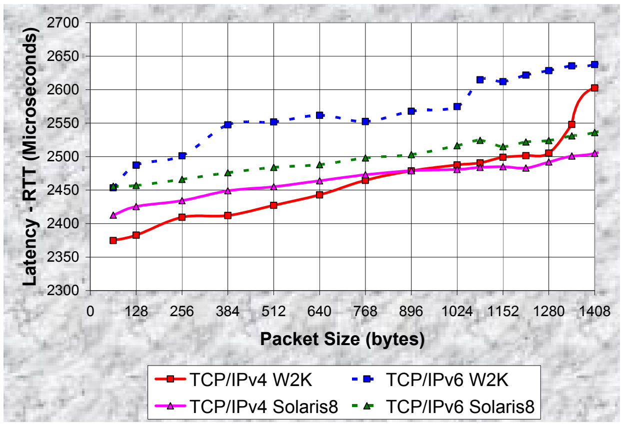

Figure 9: IBM-Ericsson Testbed: TCP latency results with packet size ranging from 64 bytes to 1408 bytes

Figure 9 shows that the latency for the smallest packet size tested revolved around 2,400 microseconds instead of about 300 microseconds for the P2P Testbed [14]. This is obviously the extra overhead that the two routers (IBM and Ericsson) are incurring; according to our results, the overhead incurred on each packet by the combination of both routers is on the order of about 2 milliseconds.

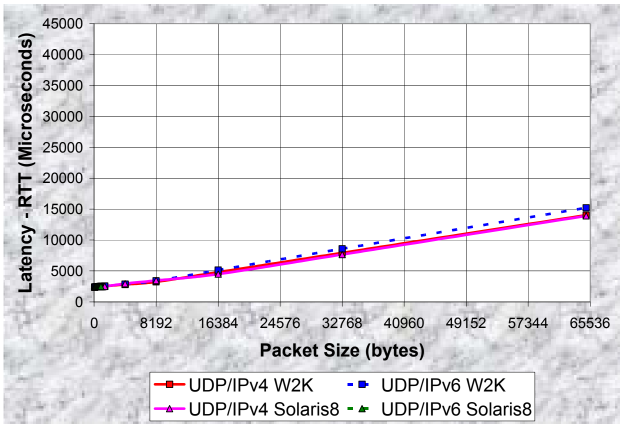

Figure 10: IBM-Ericsson Testbed: UDP latency results

In this experiment (Figure 10), we did not observe any inconsistent results. The noteworthy fact is that IPv6 incurs 1% to 8% overhead ranging from smaller packet to larger packets, while Solaris incurs a mere 1% to 4% overhead over the same packet size range.

In Figure 11, it appears that IPv6 offers near identical performance at IPv4 under Windows 2000. Under Solaris 8.0, the same comparison has a 1% to 3% overhead.

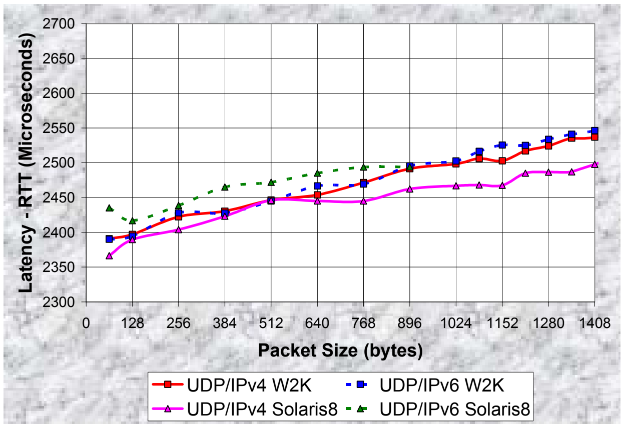

Figure 11: IBM-Ericsson Testbed: UDP latency results with packet size ranging from 64 bytes to 1408 bytes

5.2 IBM Testbed Performance Results

We have not included UDP test results for the IBM testbed because of limited space and also the results obtained were similar to those of TCP. These results that are not present in this paper can be found in [14].

5.2.1 Throughput

From Figure 12, we observe that the performance of IPv6 is around 25% lower than that of IPv4. Figure 13 above clearly shows Solaris outperforming Windows and similarly IPv4 outperforming IPv6.

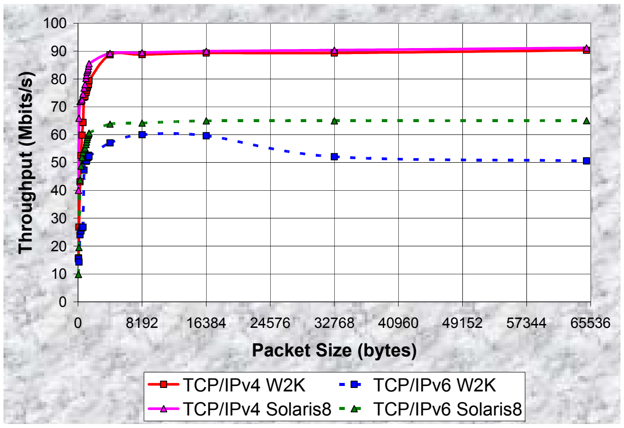

Figure 12: IBM Testbed: TCP throughput results with packet size ranging from 64 bytes to 64 Kbytes

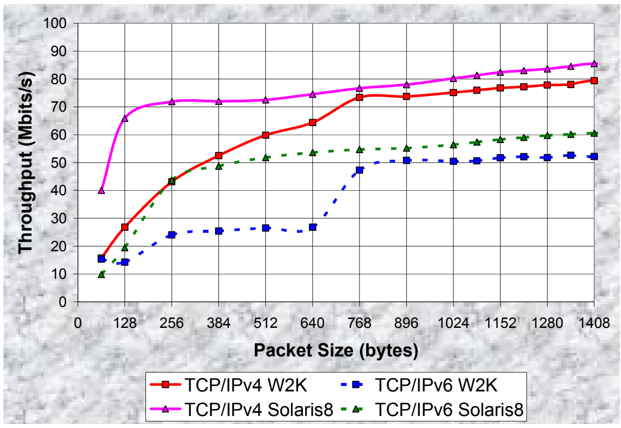

Figure 13: IBM Testbed: TCP throughput results with packet size ranging from 64 bytes to 1408 bytes

5.2.2 Latency

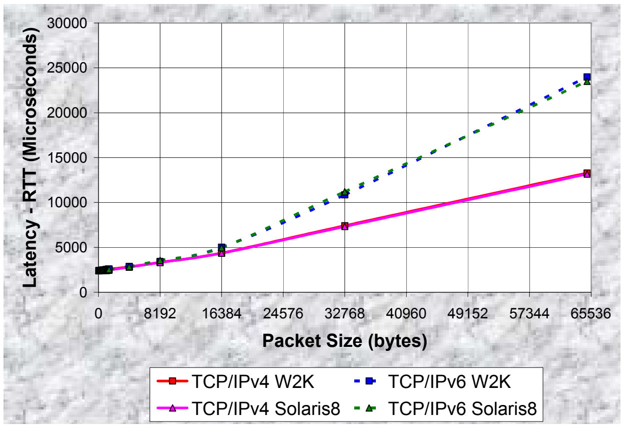

Figure 14: IBM Testbed: TCP latency results with packet size ranging from 64 bytes to 64 Kbytes

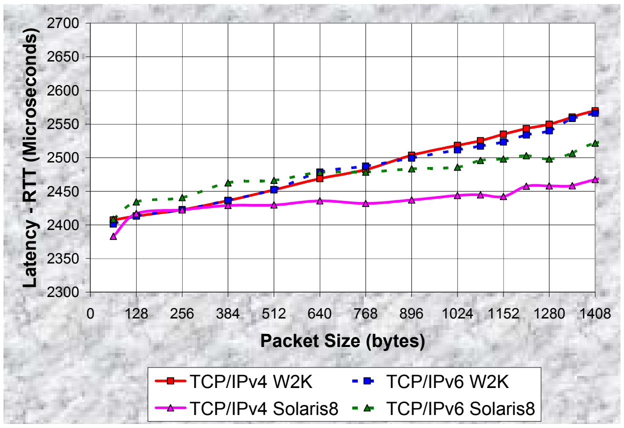

Figure 15: IBM Testbed: TCP latency results with packet size ranging from 64 bytes to 1408 bytes

From the latency tests shown in Figure 14, we observe that the RTT experiences similarly larger values for large packets. However, note that the RTT for a 64 Kbyte IPv6 packet is about 24 milliseconds while it used to be about

40 milliseconds in the IBM-Ericsson Testbed. It is clear that the IBM router is the main contributor to the increase in latency for IPv6 since the P2P configuration did not have such a large overhead [14].

5.3 Ericsson Testbed Performance Results

We repeated the experiments only the Ericsson router as shown in Figure 2. In this section, we focus mainly on the TCP Latency and throughput experiments. The UDP transport protocol's performance in the IBM-Ericsson Testbed was to be expected more or less, and therefore in order to conserve space, we will ignore them. The experiments preformed in this section use only the Ericsson router.

5.3.1 Throughput

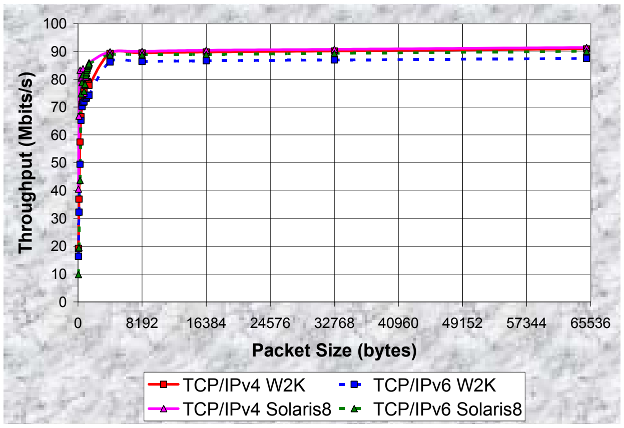

Figure 16: Ericsson Testbed: TCP throughput results with packet size ranging from 64 bytes to 64 Kbytes

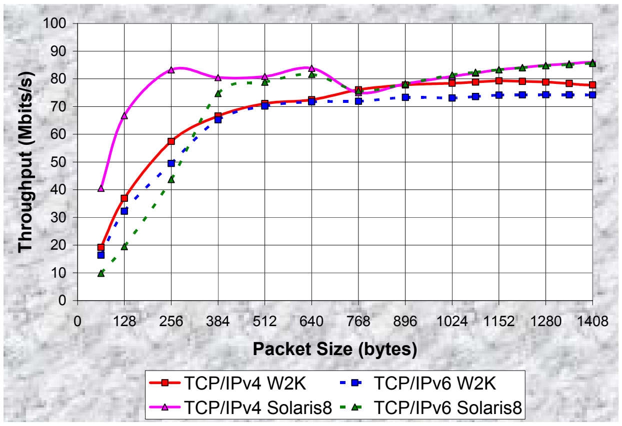

Figure 17: Ericsson Testbed: TCP throughput results with packet size ranging from 64 bytes to 1408 bytes

Both Figure 16 and Figure 17 confirm that the Ericsson Testbed has minimal impact in terms of performance overhead of IPv6 compared to IPv4. The results depicted here show that the Ericsson router handles the TCP/IPv6 packets almost as efficiently as the TCP/IPv4 packets. Obviously, there is still the usual overhead of 1% to 17% for the larger packets to the smaller ones, but this was to be expected considering the larger IPv6 header size.

5.3.2 Latency

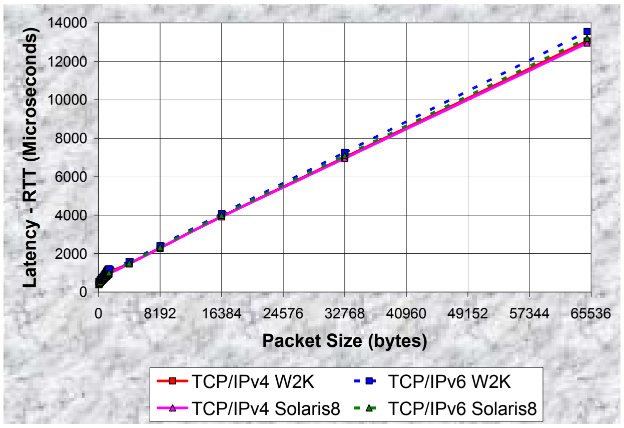

Figure 18: Ericsson Testbed: TCP latency results with packet size ranging from 64 bytes to 64 Kbytes

Figure 18 below shows that the latency incurred on the Ericsson Testbed is minimal. In Figure 19, the overheads of IPv6 increase to as much as 36% for small packets and as little as 13% for the larger packets under Windows; for Solaris, it is 5% to 7% ranging from the large packets to the small packets.

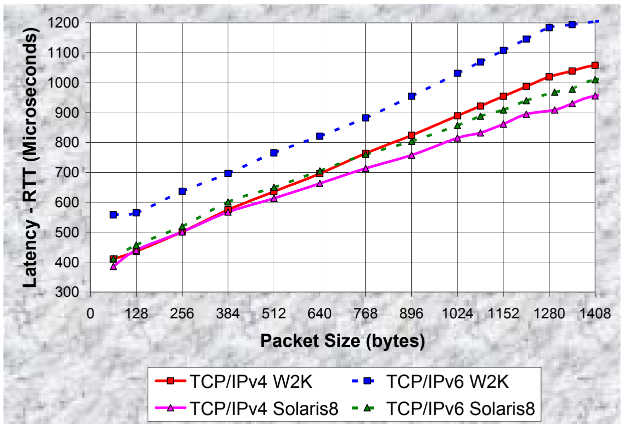

Figure 19: Ericsson Testbed: TCP latency results with packet size ranging from 64 bytes to 1408 bytes

6 CONCLUSIONS

In this paper, we have presented an unbiased empirical performance evaluation of IPv4 and IPv6 implementations, namely, Windows 2000 and Solaris 8.0 over three different local area network testbeds using commercially available routers supporting IPv6.

When we compared IPv6 protocol stacks on Solaris 8.0 and Windows 2000, we found that Solaris consistently outperform Windows in all tests. The experimental IPv6 results obtained when using different testbeds (the IBM Testbed and the Ericsson Testbed) did not yield consistent results. In the case of the IBM router, we obtained poor performance compared to when using the Ericsson router.

We observed that the throughput performance result using the IBM router was about 28% worse throughput with TCP/IPv6 compared to when handling TCP/IPv4 packets. However, in the case of the Ericsson router, we obtained worse throughput of only about 6% with TCP/IPv6 compared to TCP/IPv6 for Solaris 8. We speculate that the poor IPv6 performance exhibited by the IBM router was probably due to an early implementation of IPv6 and as a result was not as mature as the Ericsson IPv6 router implementation (the Ericsson router is one and a half years newer than the IBM router). We conclude from these results that commercially available routers supporting IPv6 are not mature yet and achieving high-end-to-end performance with the deployment of IPv6 still remains a challenge in complex heterogeneous network environments (with different operating systems, router implementations and so on).

REFERENCES

[1] Information Sciences Institute, University of Southern California, "Internet Protocol," Request for Comments 791, Internet Engineering Task Force, September 1981

[2] A. S. Tanenbaum, Computer Networks, Third Edition, Prentice Hall Inc., 1996, pp. 686, 413-449

[3] S. Bradner, A. Mankin, "IP: Next Generation (IPng) White Paper Solicitation," Request for Comments 1550, Internet Engineering Task Force, December 1993

[4] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," Request for Comments 1883, Internet Engineering Task Force, December 1995

[5] P. Srisuresh, M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," Request for Comments 2663, Internet Engineering Task Force, August 1999

[6] C. Huitema, "The H Ratio for Address Assignment Efficiency," Request for Comments 1715, Internet Engineering Task Force, November 1994

[7] R. P. Draves, et al. "Implementing IPv6 for Windows NT". Proceedings of the 2nd USENIX Windows NT Symposium, Seattle, WA, 1998

[8] P. P. Xie. "Network Protocol Performance Evaluation of IPv6 for Windows NT", Master Thesis, California Polytechnic State University, San Luis Obispo, June 1999.

[9] S. Ariga, K. Nagahashi, A. Minami, H. Esaki, J. Murai. "Performance Evaluation of Data Transmission Using IPSec over IPv6 Networks", INET 2000 Proceedings, Japan, July 18th, 2000

[10] K. K. Ettikan. "IPv6 Dual Stack Transition Technique Performance Analysis: KAME on FreeBSD as the Case", Faculty of Information Technology, Multimedia University, Jalan Multimedia, October 2000

[11] K. K., Ettikan, et al. "Application Performance Analysis in Transition Mechanism from IPv4 to IPv6". Research & Business Development Department, Faculty of Information Technology, (MMU), Jalan Multimedia, June 2001.

[12] R. Gilligan, E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers," Request for Comments 1933, Internet Engineering Task Force, April 1996

[13] Microsoft Corporation, "Microsoft IPv6 Technology Preview for Windows 2000," December 12, 2000.

[14] I. Raicu. "An Empirical Analysis of Internet Protocol version 6 (IPv6)", Master Thesis, Wayne State University, 2002.

[15] I. Raicu, S. Zeadally. "Evaluating IPv4 to IPv6 Transition Mechanisms", to appear at ICT'2003, Tahiti Papeete, French Polynesia, February 2003.