IPv4-v6 Transition Mechanisms Network
Performance Evaluation on Operating Systems

Shaneel Narayan, Sotharith Tauch
Department of Computing
Unitec Institute of Technology
Auckland, New Zealand
snarayan@unitec.ac.nz

Abstract—Last few decades has brought many fundamental changes to data communications and the Internet. Internet has its roots in a networking project started by ARPA which consisted of four computers. Now the Internet spans the world but transition to the new version has been remarkably slow. Thus in the interim, various transition mechanisms can be employed. In this paper two such mechanisms, namely configured tunnel and 6to4 transition mechanism, have been empirically evaluated for performance. Both mechanisms are implemented on two different Linux distributions, and performance related metrics like throughput, delay, jitter and CPU usage of the transition end nodes are measured. The results obtained on the test-bed show that TCP/UDP throughput and jitter values of the two mechanisms are similar, but delay reading is significantly different depending on the choice of transition mechanism and operating system.

Keywords- IPv4, IPv6, transition mechanism, configured tunnel, 6to4, network performance evaluation, Linux, Windows Server.

I. INTRODUCTION

Networks and operating systems are at the core of the information communication superhighway (the Internet). They are part of the crucial infrastructure that facilitates communication for all businesses and individuals alike. As communication requirements of business and societies change, the Internet continues to evolve to fulfill the demands. TCP/IP is the protocol suite that facilitates global Internet connectivity, and IPv4 is the currently widely used version. IPv4 has numerous limitations that are hindering the growth of the Internet, including a limited number of IP addresses. The new version of the communication protocol, IPv6, not only addresses problems that were inherent in the old version but also offers numerous opportunities to enhance user experiences on the Internet. It is expected that eventually all private networks and the Internet will be IPv6 implementations. However, the global uptake of IPv6 has been slow. In the creation of IPv6 technology, transition mechanisms were designed. These mechanisms facilitate the co-existence of IPv4 and IPv6 networks together and create communication channels between them. Numerous transition mechanisms exist and each has its strengths and weaknesses.

In this research we evaluate network performance of two transition mechanisms of various operating systems.

The rest of the paper is organized as follows: Section II discusses some similar work undertaken by other researchers, and Section III outlines the experimental setup used in this research. We present the results and discuss the findings in Section V. Finally, the research is concluded.

II. RELATED RESEARCH

There exist numerous research on topics related to IPv6 and its transition mechanisms. Three transition mechanisms, dual stacks, tunneling and translation are discussed in [1] and [2]. Poor implementations and wrong operation of IPv6 dual stack are mentioned in [3] and constraints of various transition mechanisms are discussed in [4]. The authors mention that choice of the actual mechanism is site specific. Dual Stack Transition Mechanism (DSTM) is the subject of research [5] with empirical evaluation of latency and response time of DNS traffic type on such networks. In [6], DSTM performance is tested on a test-bed. In [7] implementation of an IPv6 network on a gigabit network infrastructure is presented and the experiences discussed. A new feature in IPv6, namely network discovery is discussed in [8]. Bi-directional Mapping System (BDMS) and DSTM are compared in two scenarios in [9] [10]. 6to4 transition mechanism and tunneling are empirically compared on a test-bed setup with Windows 2000 operating system in [11]. Application Layer Gateway (ALG) for IPv6 is performance analyzed in [12]. Teredo mechanism has been evaluated using simulation is [13]. Multiple transition mechanisms have been tested for performance in [14] [15] and shown that performance degradation does occur as data traverses transition border.

In this research undertaking, two transition mechanisms, 6to4 and configured tunnel has been network performance tested on two Linux distributions and two Windows Server distributions. This is a continuation of research undertaken by authors in [16] and [17]. Typical performance related metrics are empirically measured on a test-bed setup, discussed next.
III. EXPERIMENTAL SETUP

Four computers (Intel Core 2 Duo E6300, 1.866 GHz: RAM 2GB) were connected using Cat5e cables as in Figure 1. All computers had two network interfaces (Broad-com NetXtreme Gigabit and Ethernet Intel 100s Fast Ethernet) with computers at the ends using faster NICs. Computers at the ends acted as the client nodes while computers in the middle were configured as routers. Keeping all hardware constant, operating system software on all computers was changes to Linux Fedora 9.10 or Linux Ubuntu 11.0 at a time. With each operating system, IPv4, IPv6 or IPv4/IPv6 configured tunnel and 6to4 transition mechanism was implemented and performance related metrics were measured. These metrics values were measured using D-ITG [18] which measures by using two components of D-ITG which include ITG-Send and ITG-Receive. D-ITG is capable of producing traffic at packet level for both IDT (Inter Departure Time) and PS (Packet Size). D-ITG can be used to measure throughput, packet loss, delay, and jitter analysis across heterogeneous network such as wired network, wireless network, GPRS, and Bluetooth. And in this research, we measured throughput, jitter and delay for both TCP and UDP traffic types.

To ensure high data accuracy, all tests were executed 20 times, and to get the maximum throughput for a given packet size, each run had duration of 30 seconds. The results are presented and discussed next.

IV. RESULTS AND DISCUSSION

We now present and discuss the results of this re-search. In Figure, throughput values of four operating systems are presented for TCP traffic type. Here it is seen that all operating systems portray similar behavior for both the transition mechanisms. Initially for 64Bytes packet size, all throughput values are approximately 50Mbps, and for all the other packet sizes, the values average approximately 85Mbps. A clear distinction is seen between Windows based and Linux based scenarios of a few packet specific packet sizes (640, 1280, 1408, 1536Bytes). For all these packets, Linux based scenarios have higher throughput, at most by 10%. Theoretically, maximum speed on the test bed equipment used is 100Mbps; however for both the transition mechanisms on all the operating systems, the maximum is 90Mbps, a reduction by almost 10%. UDP throughput values (Figure 3) has graphs different is appearance to the previous graph. Instead of the values increasing steeply and settling to an average of 85Mbps, the values gradually increase as packet sizes increase for most packets. Most scenarios portray similar values; however there are a few clear distinctions. Linux Fedora has the distinctively lowest values of all scenarios for smaller packet sizes (64 & 128Bytes), with all other scenarios grouping together except Windows Server 2003 with 6to4 transition mechanism. As for larger packets, Windows Server 2008 with configured tunnel and Server 2003 with 6to4 as the transition mechanism have values significantly lower than that of the other scenarios for packet sizes of 1208 and 1480Bytes – a...
reduction of almost 8%. Maximum UDP throughput value recorded is approximately 90Mbps – similar to TCP throughput maximum value.

TCP latency values in Figure 4 shows that for smaller packets, no distinction can be made between operating systems, however for larger packets (896 Bytes and above), Linux based operating system latency values are significantly lower than that of Windows based scenarios.
This difference is almost 50% for some packets. For UDP latency (Figure 5), again all scenarios have comparable values. But a clear distinction is seen between latency for large packets and the rest - for larger packets (1280Bytes and above), latency values steeply increase to almost 0.9seconds, but for most other packet sizes, the values are between 0.1 and 0.3seconds. Overall, for latency, no clear distinction can be made between the two transition mechanisms on various operating systems.

TCP delay values, shown in Figure 6, show a clear distinction between some of the scenarios. Windows Server 2008 with 6to4 and Linux Fedora with configured tunnel values are significantly higher than that of other operating systems. For all operating systems, all values are under 200ms, while for the above two mentioned scenarios, the values average 1100ms for most packet sizes. Interestingly, a similar trend exists for UDP delay values (Figure 7). Again it is seen that Windows Server 2008 with 6to4 and Linux Fedora with configured tunnel delay values are significantly higher (average of approximately 1100ms) than the rest (values less than approximately 200ms), except for Windows Server 2003. This operating system with both the transition mechanisms has similar values, however for smaller packet sizes; the values are much higher than that of larger packet sizes.

Finally CPU utilization recorded on the sending and the receiving routers are presented. For TCP traffic type on router 1, it is seen that most scenarios use resources less than 20% except for Windows Server 2008 for both the transition mechanism. It is seen that this new operating system from Microsoft, CPU utilization on router 1 is much higher than the other scenarios by almost double for some packet sizes. For packet size of 256Bytes, CPU utilization reaches almost 40%. No clear distinction between the operating systems or the transition mechanisms can be made for CPU utilization on Router 2 for TCP traffic type (Figure 9). Most values are concentrated between 10 and 20%, like on Router 1 and for Windows Server 2008; CPU utilization has decreased for most packet sizes. For UDP traffic type, CPU utilization on the two routers is shown in Figures 10 and 11. In both cases, it is seen that three scenarios group together and has graphs are distinct to the other scenarios. In these three scenarios (Windows Server 2008 with configured tunnel, Windows Server 2008 with 6to4 and Windows Server 2003 configured tunnel), higher CPU usage (up to 45% usage) is recorded for smaller packets (smaller than 384Bytes), and for all other packets, the values are comparable with the other operating systems. It is also observed that on Router 2, Windows Server 2003 with configured tunnel CPU utilization is consistently lower than the other scenarios.

V. CONCLUSIONS

In this research, we empirically evaluated performance of two transition mechanisms (configured tunnel and 6to4) on two Linux distributions (Fedora and Ubuntu) and two Windows Server operating systems (Server 2003 and 2008) by measuring four performance metrics (throughput, latency, delay and CPU usage). TCP and UDP traffic types were simulated using D-ITG on a test-bed setup. From this
empirical test-bed evaluation, the following specific conclusions can be drawn:

1. Throughput values for the four operating systems with the two transition mechanisms are comparable. There is hardly any difference in values in the four scenarios except for some packet sizes where Linux distributions outperform Windows operating systems by almost 10%.

2. Latency values all follow a similar pattern for both TCP and UDP traffic type for all four operating system with the two transition mechanism. Again there is hardly any significant difference between the operating systems or the two transition mechanisms except in one situation (TCP latency) where Linux distributions give significantly lower latency than the rest.

3. Delay experienced in all the scenarios show that Fedora with configured tunnel and Windows Server 2008 with 6to4 has significantly higher delays than the other scenarios. For both TCP and UDP, Windows Server with 6to4 and Linux Fedora with configured tunnel have delay values approximately 1100ms and the rest has less than 200ms.

4. CPU usage of the two routers show hardly any difference in values, however on Router 1, Windows Server 2008 values are slightly higher than the others.

This research has shown that the performance of transition mechanisms is generally consistent on the two operating systems tested, however one metrics in particular, average delay, is different depending on the operating system and the transition mechanism. The re-search team aims to extend this study to incorporate more operating systems including Windows operating systems.

REFERENCES


