Volume 2, Issue 10, October 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Sustainable IPv4 to IPv6 Transition

Chiranjit Dutta[1], Ranjeet Singh[2]
Department of Information Technology
SRM University, NCR Campus
India

Abstract: ***The next-generation Internet Protocol, initially known as IP Next Generation (Ipng), and then later as IPv6, has been developed by the Internet Engineering Task Force (IETF) to replace the current Internet Protocol (also known as IPv4). To enable the integration of IPv6 into current networks, several transition mechanisms have been proposed by the IETF IPng Transition Working Group. This work examines and empirically evaluates two transition mechanisms, namely IPv6 to IPv4 tunneling and dual-stack mechanism, as they relate to the performance of IPv6. We explore the impact of these approaches on end -to-end user application performance using metrics such as throughput, latency, and host CPU utilization. All experiments were conducted using three dual stack (IPv4/IPv6) routers, an IPv6 router and two ends –stations running Windows 7, loaded with a dual IPv4/IPv6 stack.***

Keywords: IPv4, IPv6, 6-over-4, encapsulation, tunneling, performance evaluation

1. Introduction

The rapid diffusion of the Internet and development of high-speed broadband networks have posed the problem of inadequate IPv4 address space on the Internet. Moreover, this lack of address space has been made worse by the progress made toward a ubiquitous network society, in which various types of information equipment, mobile computers, and electrical information appliances communicate on the Internet. IPv6 was developed as a solution to this problem [1]. The IPv4 address structure is based on a 32-bit address length. It can manage about 4 billion addresses, but cannot be assigned to everyone living in the world, which contains about 6.3 billion people. The next-generation Internet Protocol, initially known as IP Next Generation (Ipng), and then later as IPv6, has been developed by the Internet Engineering Task Force (IETF) to replace the current Internet Protocol (also known as IPv4). When both IP versions are available and the users of Internet want to connect without any restrictions , a transition mechanism is required.

To enable the integration of IPv6 into current networks, several transition mechanisms have been proposed by the IETF IPng Transition Working Group. The ubiquitous network society requires that the IPv6 network (or next-generation Internet Protocol), with its vast address space, be adapted as part of the infrastructure, because the IPv4 network currently used by the Internet cannot provide sufficient support.

However, it is not practical to reconstruct the huge user properties built into the IPv4 network for migration to the IPv6 network. Migration to IPv6 involves two technical subjects: IPv4/IPv6 translation technology and gateway construction technology. IPv4/IPv6 translation technology involves address mapping between IPv4 and IPv6, and the methods used to translate protocols. Gateway construction technology involves reserving the necessary scalability and ensuring the required reliability. IPv4/IPv6 transition always occurs in the deployment of IPv6-based services across the IPv4 Internet. This research paper mainly addresses possible enhancements and additional functionality to the proposed transition mechanisms [1, 2].

2. Background

2.1 IPv4 and IPv6 Architecture

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)or CIDR (Classless Inter Domain Routing) [3], however work began on a new Internet Protocol, namely IPv6. The main reason for a new version of the Internet Protocol was to increase the address space; IPv6 was designed with a 128 bit address scheme, enough to label every molecule on the surface of the earth with a unique address [3] . Furthermore, the only kind of traffic that existed on the internet twenty years ago was elastic traffic, such as emails or file transfers. These kinds of traffic were very flexible regardless of the network conditions; on the other hand, inelastic traffic requires a certain level of guaranteed performance, which if not met, the application does not have the same usefulness. IPv6 was designed for efficiently supporting both elastic and in elastic traffic. The goals of IPv6 were to support scalability, security, and multimedia transmissions. First, the address space is increased from 32 bits to 128 bits. The goals of IPv6 were to

support scalability, security, and multimedia transmissions. First, the address space is increased from 32 bits to 128 bits. Unlike IPv4, IPSec support has become a requirement in the IPv6 header. Payload identification for QoS handling by routers is now supported by the Flow Label field in the IPv6 packet header. Fragmentation support has been moved from routers to the sending hosts. The IPv6 header does not include a checksum and has no options included in the header, but rather introduces extension headers. Finally, IPv6 requires no manual configuration or DHCP (Dynamic Host Configuration Protocol), which will become important as the number of nodes increases [3]. Overall, IPv6 was carefully thought out and was designed with future applications in mind.

The main difference in the packet layout 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 reduced the number of required fields and made them optional as extension headers [6]. Since the Ethernet MTU size is 1514 bytes, the additional 20 bytes of header information only incur an additional 1.3% overhead; an additional 20 bytes of header information when an IPv6 packet is encapsulated in an IPv4 packet raises the overall overhead to 2.6%. In theory, this performance overhead between these two protocols is minimal.

2.2 IPv4 to IPv6 Transition mechanisms

The transition between the IPv4 Internet today and the IPv6 Internet of the future will be a long process during both protocols coexists. Figure1 shows the transition plan. A mechanism for ensuring smooth stepwise and independent changeover to IPv6 services is required. Such a mechanism must help the seamless coexistence of IPv4 and IPv6 nodes during the transition period. IETF has created the Ngtrans Group to facilitate the smooth transition from IPv4 to IPv6 services [5]. The various transition strategies can be broadly divided into three categories, including dual stack, tunneling and translation mechanisms.

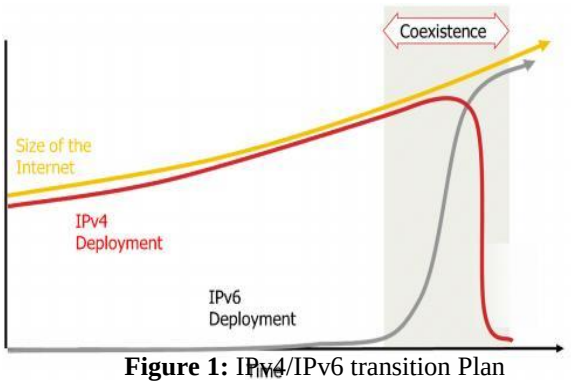

Figure 1: IPv4/IPv6 transition Plan

2.2.1 IPv4/IPv6 Dual-Stack Mechanism

The term "dual-stack" refers to TCP/IP capable devices providing support for both IPv4 and IPv6. It is important to understand that having a device being able to communicate over both IPv4 or IPv6 does not necessarily means that all applications operating within this device are capable of utilizing both IPv4 and IPv6 [5]. The term "Dual-stack routing" refers to a network that is dual IP, that is to say all routers must be able to route both IPv4 and IPv6.

Requiring all new devices be both IPv4 and IPv6 capable permits these devices to have the ability to use either IP protocol version, depending on the services available, the network availability, service, and the administrative policy (Figure 2). A transition scenario which calls for "dual-stack everywhere" provides the most flexible operational environment [5]. Dual-stacked hosts running on a dual-stack network allow applications to migrate one at a time from IPv4 transport to IPv6 transport. Legacy applications and devices that are not yet upgraded to support access to the IPv6 stack can coexist with upgraded IPv6 applications on the same network system.

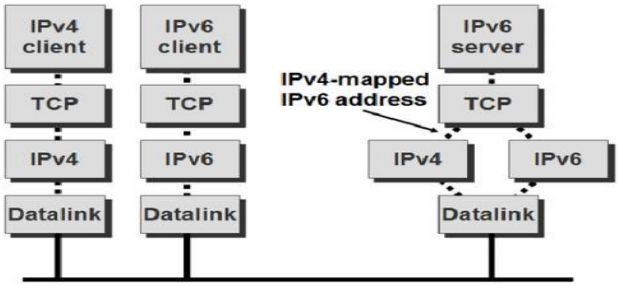

Figure 2: Dual Stack Transition Mechanism

2.2.2 IPv4/IPv6 Tunneling mechanism

The term "tunneling" refers to a means to encapsulate one version of IP in another so the packets can be sent over a backbone that does not support the encapsulated IP version. For example, when two isolated IPv6 networks need to communicate over an IPv4 network, dual stack routers at the network edges can be used to set up a tunnel which encapsulates the IPv6 packets within IPv4, allowing the IPv6 systems to communicate without having to upgrade the IPv4 network infrastructure that exists between the networks (Figure 3).

Configured Tunnels

The term "configured tunnels" is used when network administrators manually configure the tunnel within the endpoint routers at each end of the tunnel. Any changes to
the network like renumbering must be must manually reflected on the tunnel endpoint. Tunnels result in additional IP header overhead since they encapsulate IPv6 packets within IPv4 (or vice versa).

Automatic Tunnels

The term "automatic tunnels" is used when a device directly create their own tunnels to dual stacked routers for shipping IP packets within IP. The IPv6 Tunnel Broker (RFC 3053), 6to4 (RFC 3056), Teredo (Tunneling IPv6 over UDP through NATs- RFC 4380) and ISATAP (IntraSite Automatic Tunnel Addressing Protocol) ship IPv6 packets within IPv4 and can be referenced as IPv6-over IPv4 mechanisms while DSTM (Dual-stack Transition Mechanism) ships IPv4 packets within IPv6 and can be reference as IPv4-over-IPv6 mechanism [6].

The IPv6 tunnel broker mechanism uses dual-stacked servers sitting between IPv6 and IPv4 networks to assist in the setup of a configured tunnel to a host. 6to4, Teredo and ISATAP allow end host systems to create their own automatic tunnels to dual-stacked routers for shipping IPv6 packets within IPv4 [4]. While ISATAP is mainly for IPv6 -over-IPv4 tunneling within a domain, all of the other IPv6-over-IPv4 mechanisms are designed to tunnel IPv6 packets out of an IPv4-only administrative domain. Like configured tunnels, automatic tunneling has double IP header overhead, since tunnels encapsulate IPv6 packets within IPv4 (or vice versa).

DSTM technique provides a unique solution to the IPv4-IPv6 transition problem. This mechanism is designed to rapidly reduce the reliance on IPv4 routing and is intended for IPv6-only networks in which hosts still occasionally need to exchange information directly with other IPv4 hosts or applications. Network administration is simplified and the need of IPv4 global addresses is reduced. DSTM can be integrated with an IPv6 Tunnel Broker for tighter security integration. DSTM routers can be coupled with IPv4 Firewalls and Intrusion Detection systems to secure IPv4 tunnel endpoints from IPv4-based attacks.

Special consideration must be given to the security risk associated with automatic tunneling as it allows user-nodes to establish tunnels that may bypass a site's security checkpoints such as firewalls and intrusion detection. systems [7]. In general, a full dual -stack along with IPv6-capable firewalls, guards, intrusion detection, and end –host security may provide a more secure and interoperable IPv6 transition solution than tunneling. However, for network infrastructures that contain IPv4 -only or IPv6-only routing coupled with dual-stack end-nodes, automatic tunnelling provides a flexible transition strategy. Again the risks associated with all potential solutions must be carefully considered.

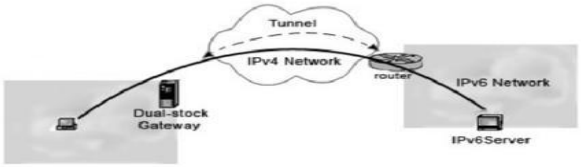

Figure 3: IPv4/IPv6 Tunneling

IPv6 Tunnel Broker

The IPv6 Tunnel Broker provides an automatic configuration service for IPv6 over IPv4 tunnels to users connected to the IPv4 Internet. IPv4 connectivity between the user and the service provider is required. The service operates as follows (Figure 4).

1. The user contacts Tunnel Broker and performs the registration procedure.
2. The user contacts Tunnel Broker again for authentication and providing configuration information (IP address, operating system, IPv6 support software, etc.).
3. Tunnel Broker configures the network side end -point, the DNS server and the user terminal.
4. The tunnel is active and the user is connected to IPv6 networks.

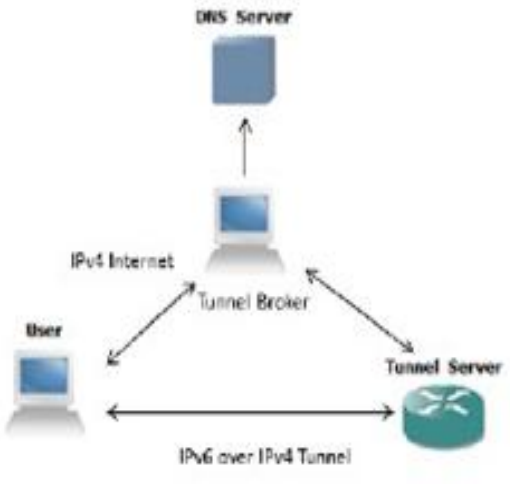

Figure 4: IPv6 Tunnel Broker

2.2.3 IPv4/IPv6 Translation mechanism

The basic function of translation in IPv4/IPv6 transition is to translate IP packets. Several translation mechanisms are based on the S2T (Stateless IP/ICMP Translation algorithm) algorithm. The S2T algorithm is used as a basis of the BIS (Bump In the Stack) and NAT-PT (Network Address Translation-Protocol Translation) mechanisms.

3. System Architecture

The main goal of this work is to measure the performance of the tunneling and dual-stack mechanisms. The performance of these mechanisms on a network, including nodes and routers that support dual IPv4/IPv6 stacks, were examined. Tunneling supports IPv6 implementation using the existing IPv4 infrastructure without changing the IPv4 modules in the early age. The entire testing process was carried out within the GNS3 Emulation Environment using a virtual topology.

3.1 Dual Stack TestBed

Dual-stack mechanisms include two protocol stacks that operate in parallel and allow network nodes to communicate either via IPv4 or IPv6. They can be implemented in both end systems and network nodes. In end systems, they enable both IPv4 and IPv6 applications to operate at the same time. The Dual-stack capabilities of network nodes support the transport of both IPv4 and IPv6 packets. In the dual-stack mechanism, specified in IETF RFC2893, a network node includes both IPv4 and IPv6 protocol stacks in parallel. IPv4 applications use the IPv4 stack, and IPv6 applications use the IPv6 stack. Flow decisions are based on the version field of IP header for receiving, and on the destination address type for sending.

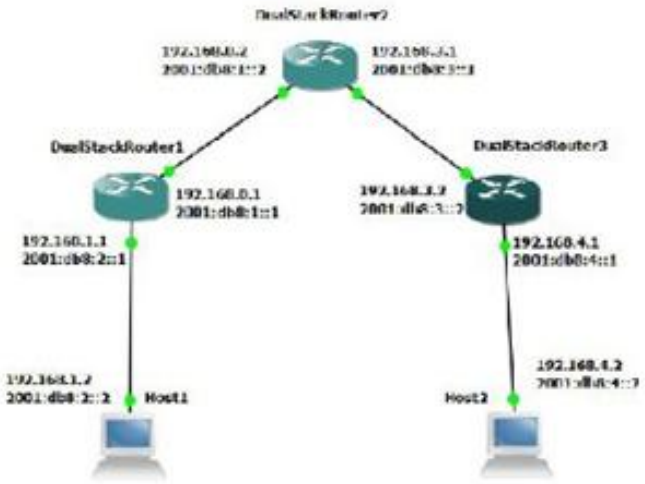

Figure 5: Dual Stack Configuration

The types of addresses are usually derived from DNS lookups; the appropriate stack is selected in response to the types of DNS records returned. Hence, the dual-stack mechanism is the most extensively employed transition solution. However, dual stack mechanisms enable only similar network nodes to communicate with each other (IPv6-IPv6 and IPv4-IPv4). Much more works are required to create a complete solution that supports IPv6-IPv4 and IPv4-IPv6 communications. Our configuration consisted of three Dual Stack Routers based on Cisco 3640 and running Cisco IOS Software. All of the routers and hosts are interconnected via Ethernet interfaces.DualStackRouter2 is connected to both DualStackRouter1 and DualStackRouter3 directly. One Host each is connected directly to DualStackRouter1 and DualStackRouter3. Each pair of router and/or hosts is on its own separate network segment. Each of the routers is running RIP for IPv4 and RIPng for IPv6 as routing processes. Each interface is assigned both IPv4 and IPv6 addresses manually according to their network segments. No static routes are defined.

3.2 Tunnel Testbed

IPv6 to IPv4 tunneling represents a mechanism for assigning an IPv6 address prefix to a network node which has a global IPv4 address. It can connect to anot-
her, by transmitting encapsulated IPv6 packets over an existing IPv4 infrastructure with minimal manual configuration.
The aim of this mechanism is to enable isolated IPv6 sites (or nodes) that attached to a native IPv4 etwork to communicate with an IPv6 domain. The IPv6 to IPv4 mechanism is implemented as a suitable tunneling behavior on border routers. Those routers are called IPv6 to IPv4 routers. A network node at an IPv6 site sends packets that are default routed to the IPv6 to IPv4 gateway and then tunnels packets to the IPv6 to IPv4 relay router. IPv6 to IPv4 relay router decapsulates packets and forwards them over the global IPv6 network with native IPv6 route. Each 6to4 network is connected to the rest of the IPv6 network via a local 6to4 gateway and a remote relay router. Packets that will transfer to IPv6 to IPv4 nodes are routed to the relay router. The IPv6 to IPv4 relay router encapsulates IPv6 packets in IPv4 packets, with destinations determined from the IPv6 to IPv4 address, and send them back. Then, the IPv6 to IPv4 gateway decapsulates the IPv6 packet and forwards it to the IPv6 site.

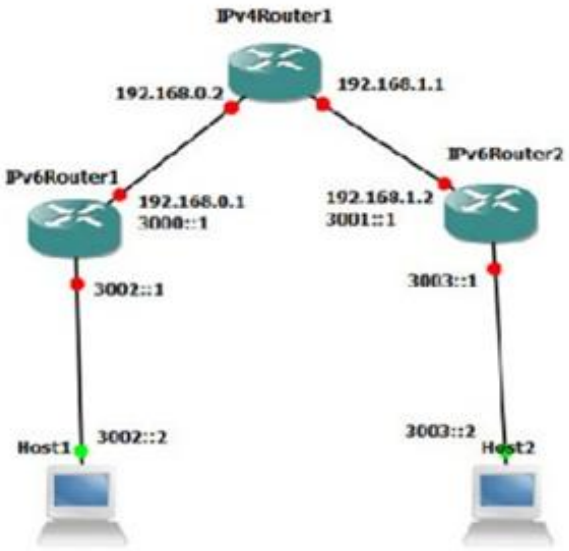

Figure 6: Tunnel Configuration

Our configuration consisted of two IPv6 routers and one IPv4 router based on Cisco 3640 and running Cisco IOS Software. All of the routers and hosts are interconnected via Ethernet interfaces. IPv4Router1 is connected to both IPv6Router1 and IPv6Router2 directly. One Host each is connected directly to IPv6Router1 and IPv6Router2. Each pair of router and/or hosts is on its own separate network segment. IPv4Router1 is assigned IPv4 addresses and is running RIP. The host-side interfaces of IPv6Router1 and IPv6Router2 are assigned only IPv6 addresses. The router-side interfaces of IPv6Router1 and IPv6Router2 are assigned both IPv6 and IPv4 addresses. Both these routers are running RIP and RIPng. A Tunnel is setup between the two router-side interfaces of IPv6Router1 and IPv6Router2. This enables IPv6 packets to travel across the IPv4 defined.

4. Performance Evaluation

4.1 Latency Analysis

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; the performance is measured in microseconds per RTT. In evaluating the performance of the tunneling and dual-stack mechanisms, the average transmission latency was measured first. Typically, the average transmission latency is the time taken for a packet to be transmitted across a network connection from sender to receiver. Tests were performed using the GNS3 program run on a reliable ICMPv6 Internet layer. It Sends ICMPv6 packets to the command argument specified network node and checks the replied message.

Latency was measured by sending different size packets, 64, 128, 256, 512, 768 and 1024 bytes. Upon the receipt of the packet, the server sent the same size packet back to the original client. When the client receives the packet, the whole process is completed. The cycle was iterated 10,000 times for a more precise result. The figure shows the comparative latency of the tested. It presents latency measured as the packet size was varied from 64 bytes to 1024 bytes. It indicates clearly that the dual stack mechanism has the least latency.

Our testing method consisted of end to end pings over the entire network. The pings were done for ICMPv6 only.

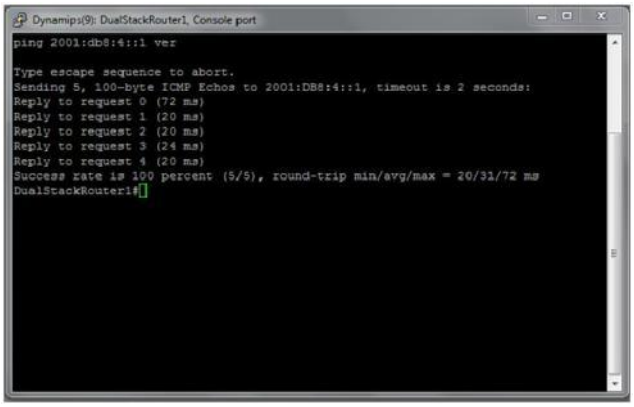

Figure 7: Latency Test

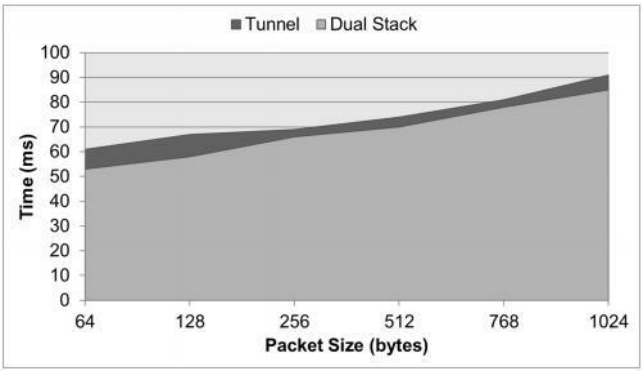

Figure 8: Latency Test Results

4.2 Throughput Analysis

Throughput is defined as the amount of packet data that is transmitted over the entire path per time unit. The throughput is calculated from the formula T=P/L where T represents the throughput, P represents the transferred data size, and L represents the time cost in transfer. The graph plots the throughput associated with the two mechanisms, for packet sizes that range from 64 bytes to 1024 bytes.

Our testing method consisted of an IPv6 compatible FTP server running on a host machine at one end of the network. Files were downloaded on a host machine at the other end of the network while throughput was being monitored. To ensure the data was moving through the network packet analysis was done using Wireshark on a router interface. The diagram clearly shows that FTP packets were indeed moving via the routers.

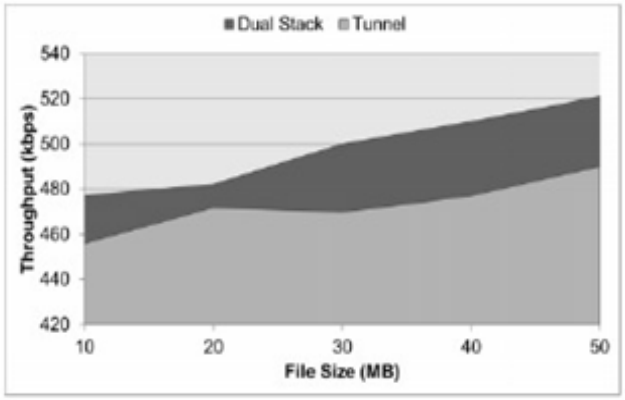

Figure 8: Throughput Test Results

4.3 CPU Usage Analysis

CPU utilization normally refers to the percentage of CPU time taken by a running process. CPU utilization at the sending node was measured using the Windows 7 Task Manager's performance monitoring tool. The average CPU utilization was recorded throughout each corresponding experiment; the performance is measured in percentage of utilization.

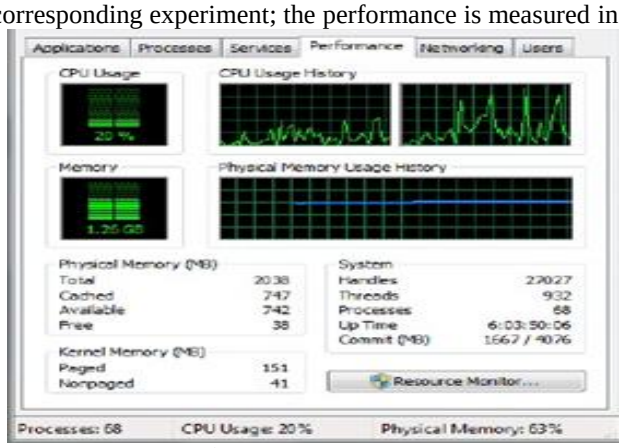

Figure 9: CPU Usage

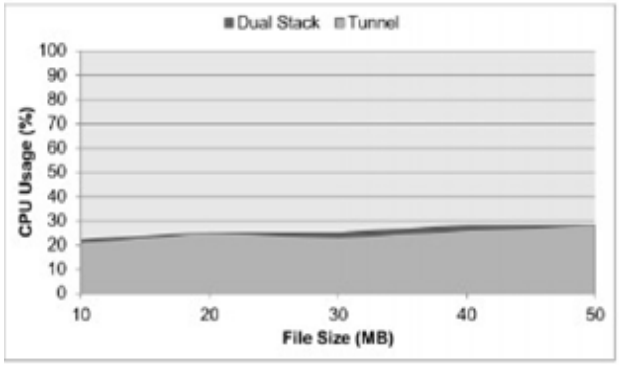

Figure 9: CPU Usage Results

5. Conclusions and Future Work

In this paper, we presented an unbiased empirical performance evaluation of IPv6, and two transition mechanisms, namely Dual Stack, and Tunneling (IPv6 in IPv4 tunneling), over a local area network testbed.

We summarize our major results presented in this work below.

- Dual Stack mechanism was found to have lower latency in all of the packet size tests. Consistency of latency was also found to be better in comparison to Tunnel mechanism
- Tunnel mechanism was found to have lower throughout values than Dual Stack which peaked at 533 kbps. As expected throughput increased as the duration of file transfer increased for larger file sizes.

- No significant deviations were found in CPU usage for both Dual Stack and Tunnel mechanisms though it was expected that Tunnel mechanism would have slightly higher CPU usage.

Future work will continue our evaluation with more transition mechanisms in the hopes to eventually empirically evaluate all the available transition mechanisms. We also intend to investigate the performance of IPv6 when exploiting IPv6 features (such as the flow label field in the IPv6 header) to investigate end-to-end QoS support in IPv6 over IP-based networks.

References

[1] Microsoft, "IPv6/IPv4 Coexistence and Migration," White Paper, Washington, November2001
[2] A. S. Tanenbaum, Computer Networks, Third Edition, Prentice Hall Inc., 1996, pp. 686, 413-436, 437-449
[3] T. Dunn, "The IPv6 Transition," IEEE Internet Computing, Vol.6, No.3, May/June 2002, pp.11-13
[4] W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols, First Edition December 15, 1993
[5] IETF IPv6 Transition Working Group, http://www.6bone.net/ngtrans.
[6] IPv6 users' site: http://www.ipv6.org
[7] IPv6 Forum, The New Internet: Internet for everyone
(www.ipv6forum.com)

