**EURASIAN JOURNAL OF MATHEMATICAL
AND COMPUTER APPLICATIONS**
ISSN 2306–6172
Volume 2, Issue 3(2014) 30–42

PERFORMANCE OF CPU UTILIZATION FOR IPv6 TUNNELING MECHANISMS ON LINUX BASED TESTBED

A.S. Wahid, M. Othman, O. Sembiyev, M.H. Selamat

Abstract

The Internet Protocol version 4 (IPv4) is showing its limitations as global communications and service demands increase and new Internet applications are developed. Internet Protocol version 6 (IPv6) is the next-generation protocol designed to replace IPv4. Transitioning from IPv4 to IPv6 is important because IPv6 provides a network infrastructure that is scalability and security. IPv6 can enable advanced applications for communications and can provide a robust foundation for the 21st century information age. In this paper, we compare the performance of CPU utilization of three tunneling mechanisms which consists of 6to4, 6rd and ISATAP against native IPv6. The experimental results determine the CPU utilization overhead for each tunneling mechanism. All the experiments were carried out on a real Linux based testbed.

Key words: Tunneling mechanism, native IPv6, 6to4, 6rd, ISATAP, Performance evaluation

AMS Mathematics Subject Classification: 68M12, 68U99, 60K30, 60K99

1 Introduction

IPv4 is the current Internet protocol that is widely use across the Internet, but in the near future, there is an issues like insufficient public IPv4 address space which not allows the growth of the Internet. Nowadays, most of devices are required to have an IP address to connect to the Internet which leads to high consumption of IP addresses. Internet Engineer Task Force (IETF) has considered this issue and proposed the new IP address known as IPv6 [3, 14, 17, 18].

IPv6 is the solution to the massive growth of the Internet due to the size of the address spaces. IPv6 addressing contains 128 bits binary value that provide 2128 addresses. In the near future the current IPv4 address will slowly migrate to IPv6 addressing. Unfortunately until today, the IPv4 are still being used because of the slow migration plan and low demand for the IPv6 addresses because users are not aware of the IPv6 existence. The transition between IPv4 and IPv6 will be a long process as they are two completely separate protocols and it is impossible to switch the entire Internet over to IPv6 over night.

IPv6 is not backward compatible with IPv4. IPv4 hosts and routers will not be able to deal directly with IPv6 traffic and vice versa. As IPv4 and IPv6 addresses will co-exist for a long time, this requires the transition and inter-operation mechanisms [1].

Migrating from IPv4 address to IPv6 address is a complicated task that cannot be done in short period of time [1, 3, 4]. The size and complexity of the Internet cause this migration task to become enormously difficult and time consuming. Next Generation Transition (NGTrans) proposed three main transition mechanisms that included dual-stack, tunneling, and translation. These solutions allow IPv4 address to be able to co-exist with IPv6 address during the

Table 1: Types of IP Tunnel Mechanisms

Types of tunnel	Description
Router to Router	IPv6/IPv4 router interconnects by IPv4 infrastructure can tunnel IPv6 packet between it
Host to Router	IPv6/IPv4 host can tunnel IPv6 packet to an intermediate IPv6/IPv4 router which is reachable over IPv4
Host to Host	IPv6/IPv4 host interconnected by IPv4 infrastructure can tunnel IPv6 packet by themselves end-to-end
Router to Host	IPv6/IPv4 router capable to tunnel IPv6 packet to its destination IPv6/IPv4 host

migration period. The design of IPv6 address shows that the new version of Internet protocol was not designed to be backward compatible with IPv4, which mean IPv4 host is only capable of sending IPv4 packets to other IPv4 hosts, and the same applies to IPv6 host, which is only capable of sending IPv6 packets to other IPv6 hosts. Interoperation is a major issue when both protocols coexist on the Internet. Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure. The IPv6 in IPv4 tunneling performs the encapsulation at the routers, and hence we will refer to it as router-to-router tunneling. As mentioned in Table 1, the router-to-router tunneling enables two entire LANs to be upgraded to IPv6 while maintaining connectivity to the rest of the Internet [11, 13, 16, 18].

2 Related Works

IPv6 has several well-known and understood benefits besides the increased address space [17, 18]. These include increased end-to-end security of systems and automated address allocation for Internet connected devices which bring to the conceptual of Internet of Things (IoT). We are in the early stages in the deployment of IPv6, with few IPv6 applications in the market and network devices needed to make trade-offs between the available IPv6 services. The initial focus is on the migration and transition techniques required for the deployment [2, 4].

Currently, IPv4 is the protocol used extensively on the Internet. All communication across the Internet relies on IPv4 protocol. In order to understand this protocol, first we need to look at the addressing scheme. The IPv4 addressing contains 4 octets and each octet represents 8 bits of a binary number. The entire address space of IPv4 contains 32 bits of binary number, which mean IPv4 has 232 addresses that are equivalent to 4,294,967,296 different addresses [14, 17].

While there are no significant requests from the end user for the migration to IPv6, the development of ubiquitous IP networks, the shift to IP-based communications and the adoption of e-business strategies across numerous other technologies are all putting pressure on the available IPv4 address space. Organizations like private and public companies use Network Address Translation (NAT) and other interim measures to overcome IPv4 address space limitations, but over time these organizations will be limited in their ability to respond to address space pressures and to take advantage of capabilities offered by IPv6.

The IPv6 was developed to increase the amount of available IP address space. By managing the IPv6 transition process early and collectively, agencies will be able to get better align and synchronize transition programmers, optimize procurement, manage programmer and technical risks and manage vulnerabilities more deliberately.

The IPv4 exhaustion problem is the key reason why we need to migrate to the IPv6. As the IP-based devices keep evolving these days, we can predict in the future, most of the devices can be controlled and managed through the Internet. In order to be recognized in the global network, each devices need to have their own public address, thus the implementation of NAT can be seen as not relevant anymore. By having IPv6 address in practice, each devices may consists more than one of IP addresses if we refer to the Internet Service Provider (ISP) practice; each end customer will have their own 216 IPv6 addresses (approximately more than 65,000 number of addresses) [10, 17, 18].

The new generation of Internet user behavior dominantly brings the need of IPv6 since nowadays, everyone are trying to host individual server at their home and that is impossible if we are still rely on IPv4. The new paradigms of broadband services provided by the ISP which not only serve their customer with high speed of Internet were also can be seen as the selling point for IPv6. The introduction of triple-play services which consists of high speed Internet, IPTV and telephony which may bring customer satisfaction instead of having normal Internet access, [3, 13].

2 IPv6 Configured Tunnel

Configured tunnel is a manually configured tunneling mechanism, which enables two or more IPv6 networks to communicate across IPv4 routing infrastructure through a tunnel. Configured tunneling defined as 6over4 tunnelling where the IPv4 tunnel endpoint address is determined by configuration information on the encapsulating node. The tunnel can be either unidirectional or bidirectional. Bidirectional configured tunnels behave as virtual point-to-point link, [4]. Figure 1 shows the implementation of configured tunnel network infrastructure.

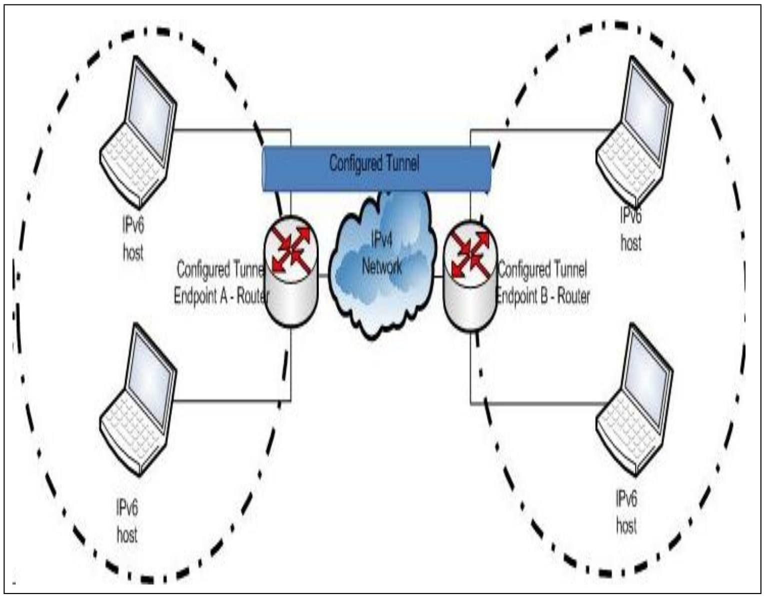

Figure 1: Configured Tunnel Network Architecture

This tunneling mechanism requires configuring each tunnel endpoint routers manually in order to deliver IPv6 across IPv4 infrastructure. Each tunnels endpoint router contains two network interface cards with the internal network interface card configured with IPv6 address and external network interface card configured with IPv4 address.

In this section, fundamental knowledge of typical configured tunnel in IPv6 tunneling mechanisms which is 6to4, 6rd and ISATAP are described, [10, 11].

2.1 6to4

The 6to4 is an address assignment and router-to-router, host-to-router, and router-to-host automatic tunnel technology that being used to provide unicast IPv6 connectivity between IPv6 sites and hosts across the IPv4 Internet [5].

As IPv6 packet arrives at 6to4 router, the encapsulation process is initiated by putting IPv6 packet in IPv4 packet in order to transmit across IPv4 Internet infrastructure. Source and destination IPv4 addresses are specified with IPv4 header. The body of IPv4 packet contains IPv6 header and payload as stated in RFC3056, [14]. The 6to4 packet is travelling across 6to4 tunneling established by 6to4 routers which also known as tunneling endpoints. As the encapsulated packet arrives at the destination tunneling end-point, 6to4 router performs de-capsulation process by removing IPv4 header and forward IPv6 packet through to IPv6 host.

The 6to4 has four components that have different functionality. Those four components are 6to4 host, 6to4 router, 6to4 host/router, and 6to4 relay. The 6to4 host is a client computer, which does not have ability to perform 6to4 tunneling across IPv4 Internet. The 6to4 router has ability to perform 6to4 tunneling across the Internet and forwarding 6to4 packet from 6to4 host in a site to another 6to4 host in another site across the Internet. The 6to4 host/router has the ability perform tunneling with 6to4 host/routers, 6to4 routers, and 6to4 relay but it does not have functionality to forward packet.

2.2 6rd

The 6rd is the extension of 6to4 that addresses this issue being name after its inventor, Remi Despres, [16]. Instead of the 6to4 2002::/16 networks, a ISP is provisioned with its own unique prefix. The Service Provider can then route the packets from the Internet. The operational domain of 6rd is inside the Service Provider's control, and thus production quality IPv6 deployment is possible. The 6rd clients need to be configured with the 6rd prefix assigned to the ISP, the length of the ISP prefix, and the 6rd relay address. The prefix may be no longer than 32 bits long. The 6rd client configures itself as *ISP Prefix + IPv4 WAN Address + Subnet ID::/64*.

The 6rd specification allows for address compression of the IPv4 WAN address. The leading bits of the address, which are unlikely to be unique within a ISP, can be dropped by specifying a compression bit mask length which leading bits to drop from the prefix.

2.3 ISATAP

The ISATAP is a method of address assignment which also provides host-to-host, host-to-router, and router-to-host automatic tunneling technology, [15, 12]. It also does provision an unicast IPv6 connectivity between IPv6 and IPv4 hosts across the IPv4 Intranet. An ISATAP host does not need any manual configuration. ISATAP addresses creation and generation are using standard IPv6 address auto-configuration mechanisms.

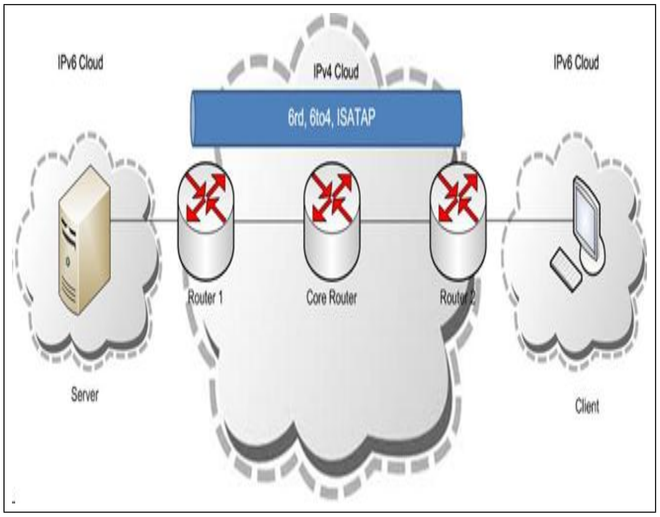

Figure 2: Testbed Architecture

The ISATAP address will automatically assign to the ISATAP interface. The ISATAP does not support router-to-router, which is the reason that this transition mechanism not selected for this experimental research study. The drawback of ISATAP is inability to implement across the Internet. The ISATAP is not designed for the Internet users, but it is design for the Intranet users.

2 Experiments Setup/Configure

In this section, telecommunication scenarios using the native IPv6, 6to4, 6rd and ISATAP tunnelling mechanisms were designed for a transition from IPv4 address to IPv6 address. The CPU utilization for each cases is measured and analysed on Linux based testbed. The rate of CPU utilization at system level for each of them is measured during the downloading large TCP file transfer protocol known as FTP protocol.

2.1 Testbed Setup

A research scenario implementation was constructed to get the real network operation scenario and output. Figure 2 shows the basic composition of the linux based testbed used in this experiment.

Testbed consists of three Linux based router running on Ubuntu 12.04 Long Term Support (LTS) kernel version 3.2.0-31. Each router will be configure accordingly to each tunneling mechanism, i.e. IPv6, 6to4, 6rd and ISATAP.

The native IPv6 setup scenarios creation is only for base lining purposes (or benchmarking purposes). Theoretically, native IPv6 give the best results among the other three tunneling mechanisms. Logically, each result from each scenario must be at par or less performed than the native IPv6 because each tunneling mechanism requires packet header encapsulation and de-capsulation, which brings to performance degradation as compared to the native IPv6 tunneling mechanism, [8].

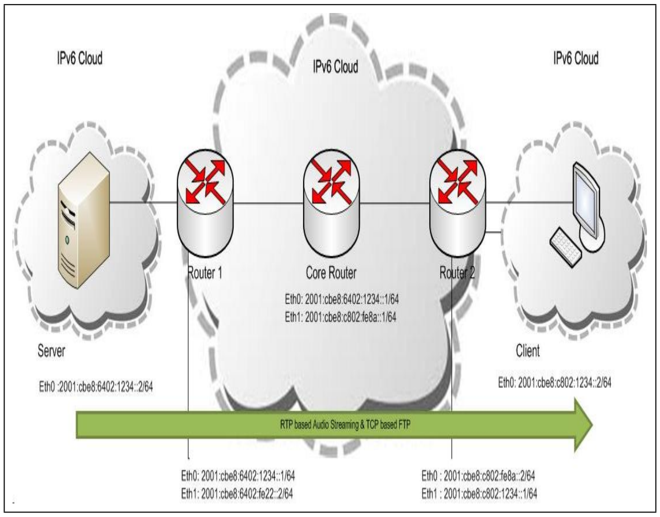
IPv6 Cloud
IPv6 Cloud
IPv6 Cloud
Router 1
Core Router
Router 2
Eth0: 2001:cbe8:6402:1234::1/64
Eth1: 2001:cbe8:c802:fe8a::1/64
Server
Eth0 :2001:cbe8:6402:1234::2/64
Client
Eth0: 2001:cbe8:c802:1234::2/64
RTP based Audio Streaming & TCP based FTP
Eth0: 2001:cbe8:6402:1234::1/64
Eth1: 2001:cbe8:6402:fe22::2/64
Eth0 : 2001:cbe8:c802:fe8a::2/64
Eth1 : 2001:cbe8:c802:1234::1/64

Figure 3: Setup for the native IPv6 testbed

2.2 Testbed Element

The research is conducted and divided into four different scenarios to achieved better results. Test-bed creations are ultimately to measure the CPU utilization for each tunneling mechanism, including the native IPv6. Figure 3 shows the setup for each tunneling mechanism's test bed.

Native IPv6 setup environment clearly state that the setup only IPv6 as their IP stack. Therefore, IPv4 configuration can be omitted. Each network devices has been configured using globally IPv6 address.

Generally, 6to4 tunneling mechanism is one of the methods to interconnect two isolated IPv6 network which have IPv4 router in between them, see Figure 4. It allocates address block normally in /64 prefixes to the host/network. The 6to4 encapsulates and wraps the IPv6 packet in the IPv4 packet and then transmitted via IPv4 using 6in4 protocol. The 6to4 are using protocol header 41 and the endpoint IPv4 address is source from the IPv6 address with the IPv6 header.

The 6rd is an automatic tunneling mechanism which being standardized by the IETF. It allows the ISPs to deploy the IPv6 as an overlay over an existing IPv4 network. By using the ISP assigned, the IPv4 address and a static set of configuration parameters received in the DHCPv4 6rd option, the dedicated home router can assign IPv6 prefixes to LAN subnets and configure a default route through a 6in4 tunnel to the ISP 6rd border router. Figure 5 shows their configuration.

The ISATAP is an IPv6 transition mechanism meant for transmit IPv6 packet between dual-stack nodes on top of an IPv4 network. The ISATAP testbed being configured as shown in Figure 6. In order to use the global ISATAP addresses, or to communicate beyond the logical subnet defined by the IPv4 Intranet, an ISATAP router is needed, [6, 7].

The ISATAP router is typically configured to perform both functions. Most often, an ISATAP router acts as the forwarder between ISATAP hosts on an IPv4 Intranet and IPv6 hosts on an IPv6 enabled portion of an Intranet, [9].

It is a method of generating a link-local IPv6 address from an IPv4 address. It is a mechanism to perform Neighbour Discovery on top of IPv4. Client who wants to participate in ISATAP over a given IPv4 network can set up a virtual IPv6 network interface. The

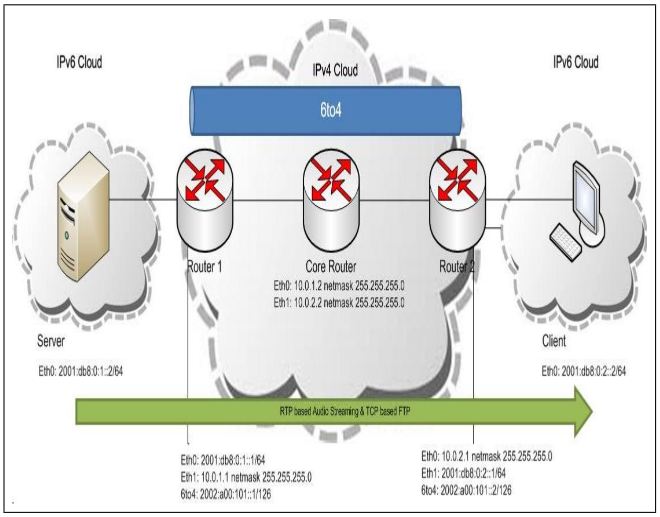
IPv6 Cloud
IPv4 Cloud
IPv6 Cloud
6to4
Router 1
Core Router
Eth0: 10.0.1.2 netmask 255.255.255.0
Eth1: 10.0.2.2 netmask 255.255.255.0
Router 2
Server
Eth0: 2001:db8:0:1::2/64
Client
Eth0: 2001:db8:0:2::2/64
RTP based Audio Streaming & TCP based FTP
Eth0: 2001:db8:0:1::1/64
Eth1: 10.0.1.1 netmask 255.255.255.0
6to4: 2002:a00:101::1/126
Eth0: 10.0.2.1 netmask 255.255.255.0
Eth1: 2001:db8:0:2::1/64
6to4: 2002:a00:101::2/126

Figure 4: Setup for the 6to4 testbed

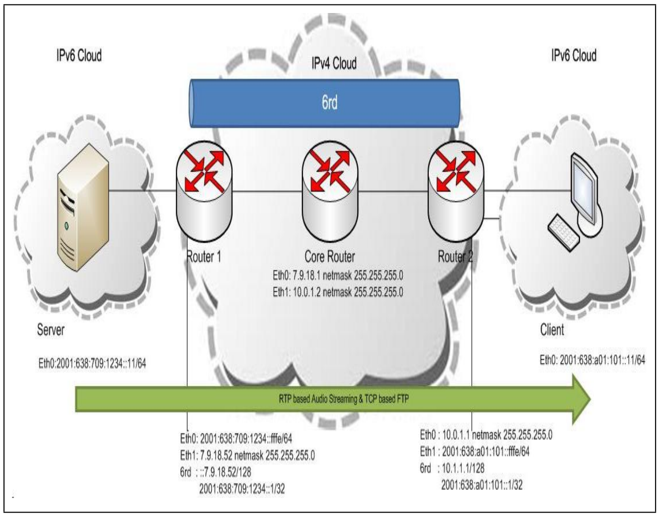
IPv6 Cloud
IPv4 Cloud
IPv6 Cloud
6rd
Router 1
Core Router
Eth0: 7.9.18.1 netmask 255.255.255.0
Eth1: 10.0.1.2 netmask 255.255.255.0
Router 2
Server
Eth0:2001:638:709:1234::11/64
Client
Eth0: 2001:638:a01:101::11/64
RTP based Audio Streaming & TCP based FTP
Eth0: 2001:638:709:1234::fffe/64
Eth1: 7.9.18.52 netmask 255.255.255.0
6rd : ::7.9.18.52/128
2001:638:709:1234::1/32
Eth0 : 10.0.1.1 netmask 255.255.255.0
Eth1 : 2001:638:a01:101::fffe/64
6rd : 10.1.1.1/128
2001:638:a01:101::1/32

Figure 5: Setup for the 6rd testbed

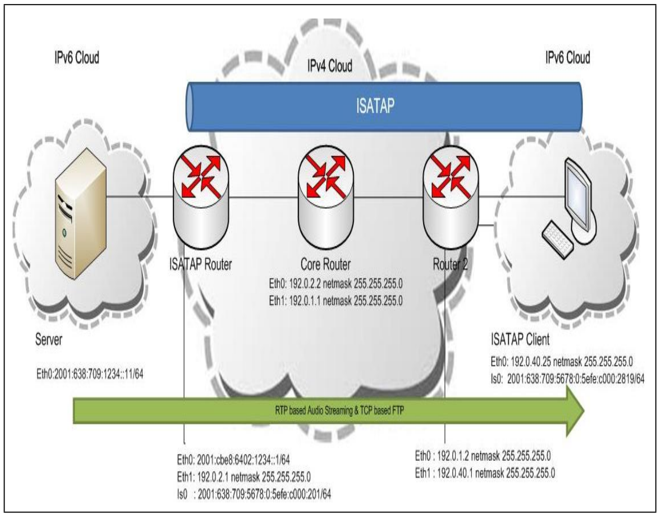
IPv6 Cloud
IPv4 Cloud
IPv6 Cloud
ISATAP
ISATAP Router
Core Router
Eth0: 192.0.2.2 netmask 255.255.255.0
Eth1: 192.0.1.1 netmask 255.255.255.0
Router 2
Server
Eth0:2001:638:709:1234::11/64
ISATAP Client
Eth0: 192.0.40.25 netmask 255.255.255.0
Is0: 2001:638:709:5678:0:5efe:c000:2819/64
RTP based Audio Streaming & TCP based FTP
Eth0: 2001:cbe8:6402:1234::1/64
Eth1: 192.0.2.1 netmask 255.255.255.0
Is0 : 2001:638:709:5678:0:5efe:c000:201/64
Eth0 : 192.0.1.2 netmask 255.255.255.0
Eth1 : 192.0.40.1 netmask 255.255.255.0

Figure 6: Setup for the ISATAP testbed

link-local address is determined by concatenating

$$fe80:0000:0000:0000:0200:5efe:$$

for global unique address and

$$fe80:0000:0000:0000:0000:5efe:$$

for private addresses with the 32 bits of IPv4 address. ISATAP router performs two main functions to advertises its presence. Address prefixes enables global ISATAP addresses to be configured and it is an optional to forward IPv6 packets between ISATAP hosts on the IPv4 Intranet and beyond the IPv6 host, [9].

2 Experimental Results and Discussions

The research aims to describe the principle of transitional mechanism between IPv4 and IPv6, make the comparison and analysis of its implementation by conducting testbed and experiment based on TCP data packets [11]. Based on [10], few of the tunneling mechanism has been tested for their performance and it is proofed that there is an overhead in each tunneling mechanism and as a result, the performance degradation do exist when the data traverse in between the routers. Figure 7 shows a result of native IPv6 vs 6to4 during the FTP session of 800Mb size of data. The CPU utilization for both mechanisms is not regular because of the changing of IP datagram sizes. It is assume that the size packet datagram for 6to4 is smaller than the native IPv6.

In Figure 8, the results show the native IPv6 vs 6rd during the FTP session. Compare with 6to4, 6rd is more fair utilization during the transmission but the utilization for both mechanisms is still not regular. Instead of having differences in the size of IP datagram, the percentage of CPU utilization is being affected by the complexity of the tunneling mechanism itself.

During the tunnel establishment, it consists of packet encapsulation and de-capsulation process throughout the data transmission.

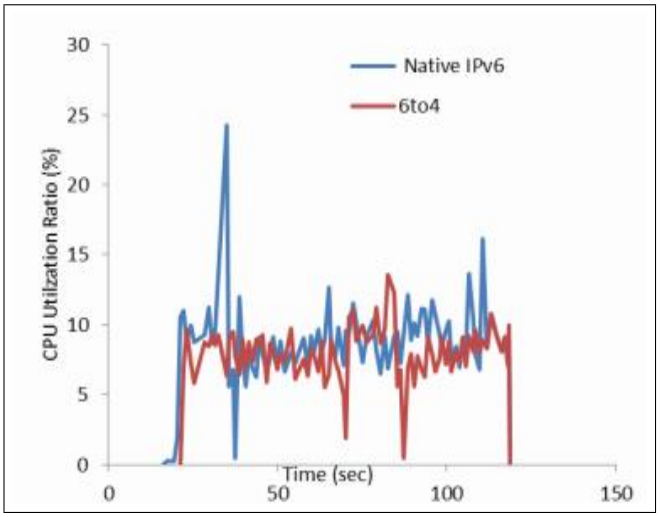

Figure 7: CPU Utilization between native IPv6 vs 6to4

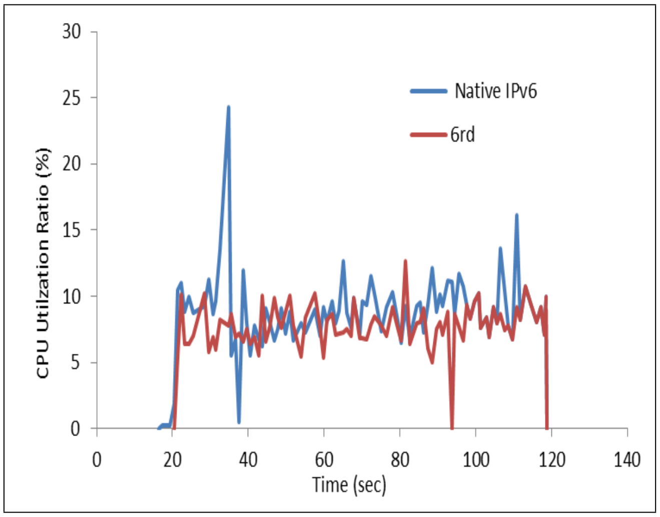

Figure 8: CPU Utilization between native IPv6 vs 6rd

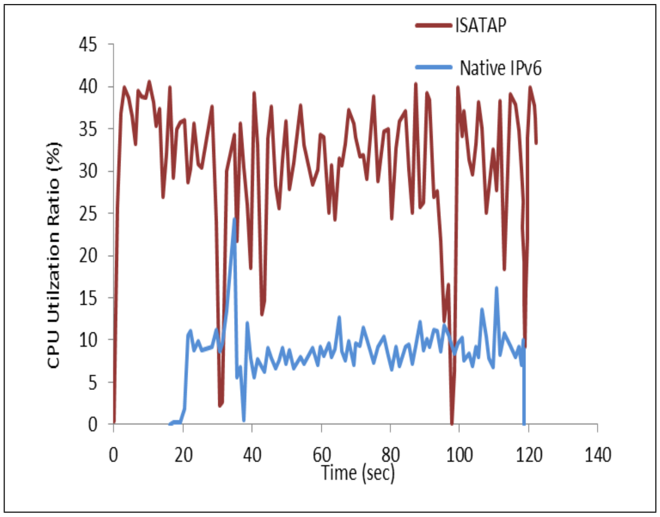

Figure 9: CPU Utilization between native IPv6 vs ISATAP

In Figure 9, it shows the result between native IPv6 vs ISATAP tunneling mechanism. It is clearly shown that the implementation of ISATAP consumed high percentage of CPU utilization during the transmission session. During the experiment, the IP datagrams sizes for ISATAP are rapidly changed throughout the period. It is also being affected by the excessive loading state which consists of massive amount of IP datagrams are received at the endpoint tunnel. Other than that, it also caused by the encapsulated packets sent by the ISATAP driver which require IPv4 host based to perform receiving windows size fragmentation in order to fulfil the 1280 bytes of IPv6 Minimum Transmission Unit (MTU).

During the packet traverse via the ISATAP tunnel, as the more time passes the more IP data received, the CPU itself may experience an excessive overload. At this time, when the data treatment for an application process is delay at a buffer, the receiving buffer senses the overflow state and executes a window update operation. This can be seen from the fluctuation of throughout. The big gap of fluctuations may be derived from the TCP based congestion control mechanism.

For both 6rd and 6to4 tunneling mechanisms, it consumes CPU processing power which more or less is the same among them. This is due to the packet encapsulation and decapsulation were occurred at the edge router. In which the result is being captured and still using IPv4 packet.

As for native IPv6 implementation, it is proof that this kind of implementation does consume a bit higher CPU utilization compared to 6rd and 6to4 because by using native IPv6 throughout the communication, it is the IP packet size which totally difference among them. In IPv4, the packet size in only size of 576 bytes which fragmentation is optional while in IPv6, the packet size is 1280 bytes without fragmentation.

From the experiment result, it is clearly shown that in the implementation of IPv6 tunneling, both 6to4 and 6rd tunneling mechanisms perform better compared to ISATAP regardless with the implementation of native IPv6. The ISATAP mechanism shows a remarkably high performance compared to the native IPv6 as the size of the datagram that should be received at the hosts reaches 1410 bytes, since 20 bytes are additionally transmitted for IPv4, which causes an overhead of 20 bytes per packet. This is in contrast to the native IPv6 system in which the IP datagram size is 1390 bytes.

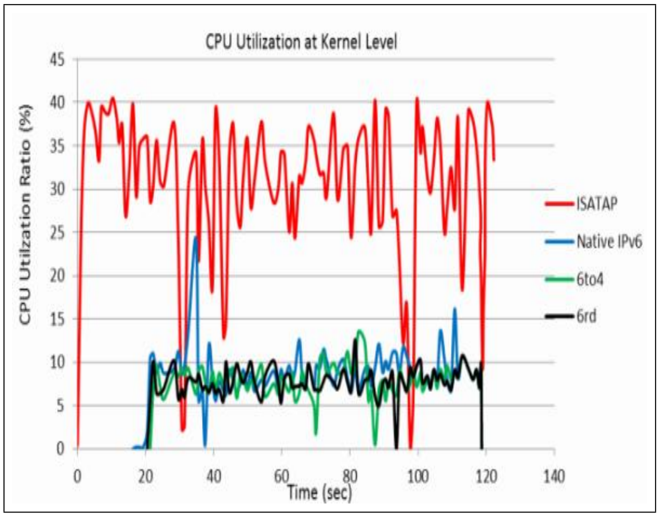

Figure 10: CPU Utilization between Native IPv6 vs 6to4, 6rd and ISATAP

The datagram size of 1410 bytes in the ISATAP method is nearly the limit that can be transmitted without packet fragmentation. For the 6to4 and 6rd mechanisms, although they also have an overhead of 20 bytes, as the datagrams are first de-capsulated at the hosts before being transmitted, the performance of these two mechanisms does not fall much as their throughput are slightly lower compared to native IPv6.

From Figure 10, comparison has been made between three tunneling mechanisms with IPv6 native setup and it is shown that the ISATAP tunneling mechanism makes the greatest CPU utilization compared to the other tunneling mechanisms. Theoretically, this is due to the large number of IP datagram that need to be de-capsulated. It may also due to the excessive overload in the IP packet itself when there is a jumbo packet being delayed at the buffer which resulting a buffer flow state. As mentioned earlier both 6rd and 6to4 transition mechanism consume CPU processing power which more or less is the same among them. This is due to the packet encapsulation and de-capsulation was done at the edge router. For native IPv6 implementation, it is proof that this kind of implementation does consume a bit higher CPU utilization compared to 6rd and 6to4 because by using native IPv6 throughout the communication, it is the IP packet size which totally difference among them. In IPv4, the packet size in only size of 576 bytes which fragmentation is optional while in IPv6, the packet size is 1280 bytes without fragmentation.

2 Conclusion

Many research papers have discussed on the tunnelling mechanism and their effects of transition from IPv4 to IPv6. In this paper, it is an experimental base research, which focused on three tunneling mechanisms which consists of 6rd, 6to4 and ISATAP. Each tunneling mechanisms were implemented on the Linux based testbed as per the experimental design and setup. Based on the experimental results collected, it is proven that for each implementation, it has its own impact. Conclusively, it was determined that both 6to4 and 6rd mechanisms are faster and more stable than the ISATAP mechanism especially for large TCP based file transfer. The ISATAP mechanism own the highest CPU utilization of its implementation with the average value of 34.2% while 6rd and 6to4 present quite similar utilization 8.8% and

8.5%, respectively.

As a future work, we plan to implement those experiments on the bigger scale of network topology of the Malaysia Research and Education Network (MyREN). We also plan to add the Quality of Service mechanism such as policing at the ingress router of the DiffServ domain with real time data traffics.

Acknowledgement

This work was supported by the Ministry of Higher Education of Malaysia under the Fundamental Research Grant FRGS/02/01/12/1143/FR for financial support.

References

[1] Sailan, M.K.; Hassan, R., *A comparative review of IPv4 and IPv6 for research test bed*, Proceeding of the International Conference on Electrical Engineering and Informatics, (2009), 427-433.

[2] Govil, J., Kaur, N., Kaur, H., *An examination of IPv4 and IPv6 networks: Constraints and various transition mechanisms*, Proceeding of the IEEE SoutheastCon, (2008), 178-185.

[3] Zimu, L., Wei, P., Yujun, L., *An Innovative IPv4-IPv6 Transition Way for Internet Service Provider*, Proceeding of the IEEE Symposium on Robotics and Applications, (2012), 672-675.

[4] Gilligan, R., Nordmark, E., *IPv6 Transition Working Group-Standard Track*, RFC2893, (2008), http://www.ietf.org/rfc/rfc2893.txt.

[5] Risdianto, A.C.; Rumani, R., *IPv6 Tunnel Broker implementation and analysis for IPv6 and IPv4 interconnection*, International Conference Telecommunication Systems, Services,and Applications, (2011), 139-144.

[6] Se-Joon Yoon, Jong-Tak Park, Dae-In Choi, Hyun K. Kahng, *Performance Comparison of 6to4, 6rd and ISATAP Tunneling Method in Real TestBed*, International Journal on Internet and Distributed Computing, vol. 2, (2012), 149-156.

[7] Sang-Do Lee, Myung-Ki Shin, Hyoung-Jun Kim, *Implementation of ISATAP Router*, Proceedings of the 8th International Conference of Advanced Communication Technology, (2006), 1160-1163.

[8] Shin, M., Kim, H., Santay, D., Montgomery, D., *An Empirical Analysis of IPv6 transition mechanism*, Proceedings of the 8th International Conference on Advanced Communication Technology, vol.3, (2006), 1990-1996.

[9] Shubhangi, K., Biranale, B.D., *IPv4 to IPv6 Transition Using Windows OS*, Proceedings of the SPIT-IEEE Colloquium and International Conference, (2008), 115-120.

[10] Chen, E., Teo, T., Isaac B., Ting, N., *Analysis of IPv6 Network Communication using Simulation*, Proceedings of the 4th Student Conference on Research and Development, (2006), 11-15.

[11] Zhi-yi, Y., Xiao-yan, L., *Study and implementation of IPv4/IPv6 transition technology based on multi-core*, Journal of Computer Applications, vol: 3, (2009) (abstract).

[12] Mohammad Azam, *Comparison of IPv6 Tunneled Traffic of Teredo and ISATAP over Real Testbed Setup*, Proceedings of 2nd IEEE International Conference on Information and Emerging Technologies, (2010), 1-4.

[13] Dutta, C., Singh, R., *Sustainable IPv4 to IPv6 Transition*, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 10, (2012), 298-305.

[14] Carpenter, B., Moore, K., *Connection of IPv6 Domains via IPv4 Clouds*, RFC3056, (2001), http://www.ietf.org/rfc/rfc3056.txt

[15] Templin, F., Gleeson, T., Thaler, D., *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*, RFC5214, (2008), http://www.ietf.org/rfc/rfc5214.txt

[16] Despres, R., *IPv6 Rapid Deployment on IPv4 Infrastructures*, RFC5569, (2010), http://tools.ietf.org/pdf/rfc5569.pdf

[17] Hinden, R., Deering, S., *Internet Protocol Version 6 (IPv6) Addressing Architecture*, RFC3513, (2003), http://tools.ietf.org/pdf/rfc3513.pdf

[18] Ibrahim Al-Surmi, Mohamed Othman, Borhanuddin Mohd Ali, *Mobility Management for IP-based Next Generation Mobile Networks: Review, Challenge and Perspective*, Journal of Network and Computer Applications, vol. 35, no. 1, (2012), 295-315

Mohamed Othman*, Ahmad Syamil Wahid, Mohd Hasan Selamat
Department of Communication Technology and Networks
Universiti Putra Malaysia, 43400, UPM, Serdang, Selangor D.E, Malaysia,
Email: mothman@upm.edu.my, syamil@tmrnd.com.my, hasan@upm.edu.my
*Corresponding author

Ordabay Sembiyev
Department of Computer Science and Software
M.O.Auezov South Kazakhstan State University, Tauke Khan,
Shymkent, Kazakhstan
Email: ordabai@mail.ru

Received 27.06.2014, Accepted 11.12.2014