European Journal of Scientific Research
ISSN 1450-216X Vol.34 No.1 (2009), pp.110-124
© EuroJournals Publishing, Inc. 2009
http://www.eurojournals.com/ejsr.htm

IPv4/IPv6 Transition Mechanisms

D.Shalini Punithavathani
Registrar, Anna University Tirunelveli, India

K.Sankaranarayanan
Dean of Electrical Sciences, V.L.B.Janakiammal College of Engineering and Technology, India

Abstract

Currently, the Internet consists of native IPv4 (IPv4-only), native IPv6, and IPv4/IPv6 dual networks. Unfortunately, IPv4 and IPv6 are incompatible protocols. When both IP versions are available and the users of Internet want to connect without any restrictions, a transition mechanism is required. During the time of migration from IPv4 to IPv6 networks, a number of transition mechanisms have been proposed by IETF to ensure smooth, stepwise and independent changeover. IPv4/IPv6 transition always occurs process in deploying IPv6-based services across the IPv4 Internet. The IETF Next Generation Transition Working Group (NGtrans) has proposed many transitions Mechanisms to enable the seamless integration of IPv6 facilities into current Networks. This Work Mainly Addresses The performance of the various tunneling transition mechanisms used in different networks. The effect of these mechanisms on the performance of end-to-end applications is explored using metrics such as transmission latency, throughput, CPU utilization and packet loss. The measured latency and throughput of the ipv6 to ipv4 mechanism are better than those of the configured tunnel and tunnel broker mechanisms the ipv6 to ipv4 mechanism must work much harder (greater overhead) for each packet sent, and it must therefore run at a higher CPU utilization of the edge router. Larger packets had higher loss rates, for all three tunneling mechanisms.

1. Introduction

The development of the IPv6 protocol, as well as being fundamental to the growth of Internet, is the basis of the increase in IP functionality and performance [1,2]. The IPv6 protocol is intentionally designed to minimize impact on layering protocols by avoiding the random addition of new features. It will support the deployment of new applications over the Internet, and open up a broad field of technological development [3,4]. Companies such as Microsoft and Nokia have issued white papers on accelerating the IPv6 progress [5, 6]. Many new applications and operation systems, including Windows XP and Linux Kernel 2.1.8 and over, already integrate IPv6 functions. But some major challenges remain before an effective and smooth transition from IPv4 to IPv6 can be ensured [7].

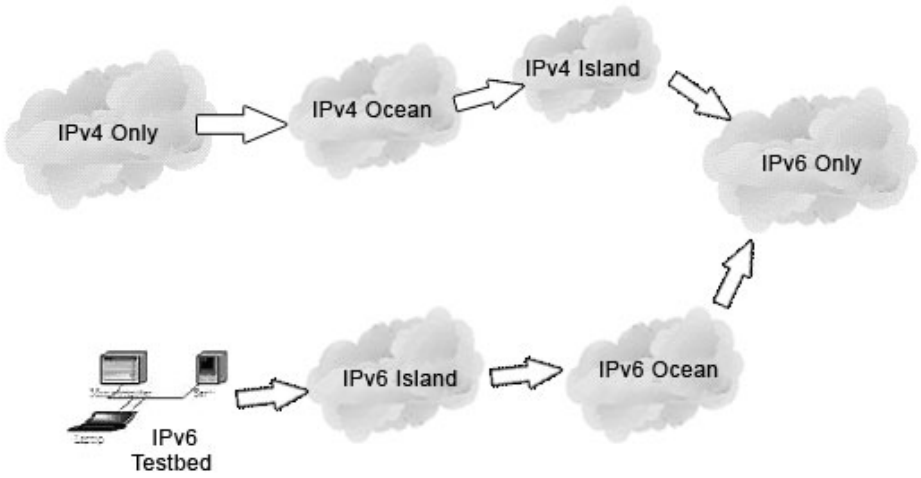
IPv6-based networks have been implemented in isolation in the past. But now, people are seeking the way to connect these IPv6 islands across the IPv4 ocean. Much of this works involve a return to simplicity and ease of use with as little disruption the existing networks as possible. Three main transition mechanisms proposed by Ngtrans [8] have already emerged, including dual stack, tunneling and translation. This work demonstrates how tunneling mechanisms could be used to

establish transparently hybrid communications between the IPv4/IPv6 worlds. Performance issues, like transmission latency, throughput, CPU utilization and packet loss, are also discussed.

2. IPv4/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. Figure 1 shows the transition phases. 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. The various transition strategies can be broadly divided into three categories, including dual stack, tunneling and translation mechanisms [9]

Figure 1: IPv4/IPv6 Transition Phases



2.1. IPv4/IPv6 Dual-Stack Mechanism

As the word means, dual-stack mechanisms include two protocol stacks that operate in parallel and allow network nodes to communicate either via IPv4 or IPv6 [10]. They can be implemented in both end system and network node. 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 (Figure 2) [11].

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. The types of addresses are usually derived from DNS lookups; the appropriate stack is selected in response to the types of DNS records returned.

Many off-the-shelf commercial operating systems already have dual IP protocol stacks [12]. 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.

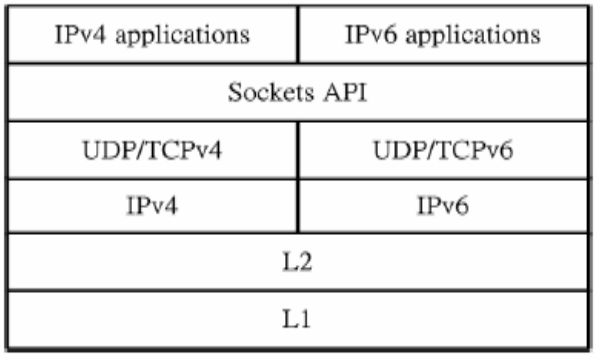
2.2. IPv4/IPv6 Tunneling Mechanisms

Tunneling, from the perspective of transitioning, enables incompatible networks to be bridged, and is usually applied in a point-to-point or sequential manner. Three mechanisms of tunneling are presented: IPv6 over IPv4, IPv6 to IPv4 automatic tunneling, and Tunnel Broker.

2.2.1. IP6 over IPv4 Mechanism

The IPv*6 over IPv4* mechanism embeds an *IPv4* address in an *IPv6* address link layer identifier part, as shown in Figure 3 and defines Neighbor Discovery (ND) over *IPv4* using organization-local multicast [13]. An IPv4 domain is a fully interconnected set of IPv4 subnets, within the scope of a single local multicast, in which at least two IPv6 nodes are present. The IPv*6 over IPv4* tunneling setup provides a solution for IPv6 nodes that are scattered throughout the base

Figure 2: Dual-stacks Transition Mechanism



IPv4 domain without direct IPv6 connectivity. The mechanism allows nodes, on physical links, which are directly connected IPv6 routers to become fully functional IPv6 nodes.

2.2.2. IPv6 to IPv4 Automatic Tunneling Mechanism

Automatic tunneling refers to a tunnel configuration that does not need direct management. An automatic IPv*6 to IPv4* tunnel enables an isolated IPv6 domain to be connected over an IPv4 network and then to a remote IPv6 networks. Such a tunnel treats the IPv4 infrastructure as a virtual non-broadcast link, so the IPv4 address embedded in the IPv6 address is used to find the other end of the tunnel. The embedded IPv4 address can easily be extracted and the whole IPv6 packet delivered over the IPv4 network, encapsulated in an IPv4 packet. No configured tunnels are required to send packets among *6to4*- capable IPv6 sites anywhere in IPv4 Internet.

Figure 4 shows the structure of the *6to4* address format. The value of the prefix field (FP) is 0x001, which the identifies global unicast address. The Top-Level Aggregation identifier field (TLA) is assigned by the IANA for the IPv*6 to IPv4* mechanism. Hence, the IPv6 address prefix is 2002::/16 and the 32 bits after 2002::/16 represent the IPv4 address of the gateway machine of the network in question. The packets thus know the way to any other network. The *6to4* mechanism is the most widely extensively used automatic tunneling technique [14]. It includes a mechanism for assigning an IPv6 address prefix to a network node with a global IPv4 address.

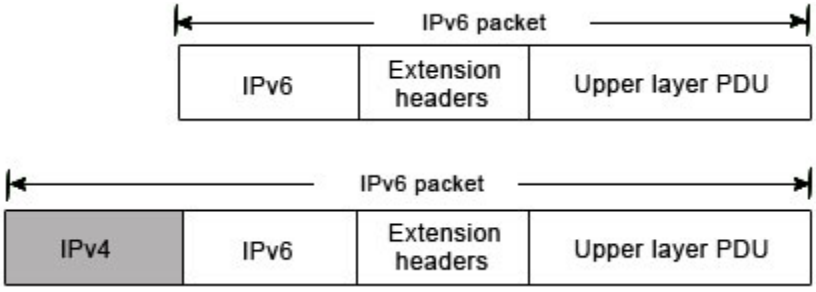
2.2.3. IPv6 Tunnel Broker

The IPv6 Tunnel Broker provides an automatic configuration service for IPv6 over IPv4 tunnels to users connected to the IPv4 Internet [15]. IPv4 connectivity between the user and the service provider is required. The service operates as follows (Figure 5).

I. The user contacts Tunnel Broker and performs the registration procedure.
II. The user contacts Tunnel Broker again for authentication and providing configuration information (IP address, operating system, IPv6 support software, etc.).

III. Tunnel Broker configures the network side end-point, the DNS server and the user terminal.
IV. The tunnel is active and the user is connected to IPv6 networks.

Figure 3: *6over4* Address Link Layer Identifier



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 SIIT (Stateless IP/ICMP Translation algorithm) algorithm [16]. The SIIT algorithm is used as a basis of the BIS (Bump In the Stack) and NAT-PT (Network Address Translation-Protocol Translation) mechanisms,

2.3.1. Bump-In-the-Stack Mechanism

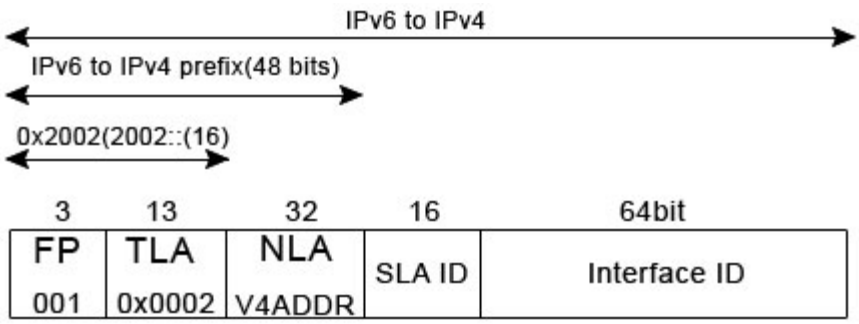
BIS mechanism (RFC 2767) includes a TCP/IPv4 protocol module and a translator module, which consists of three bump components and is layered above an IPv6 module (Figure 6) [17]. Packets from IPv4 applications flow into the TCP/IPv4 protocol module. The identified packets are translated into IPv6 packets and then forwarded to the IPv6 protocol module. The three bump components are the extension name resolver, which examines DNS lookups to determine whether the peer node is IPv6-only; the address mapper, which allocates a temporary IPv4 address to the IPv6 peer and caches the address mapping; and the translator, which translates packets between IPv4 and IPv6 protocol.

2.3.2. Network Address Translation-Protocol Translation

The NAT-PT mechanism is a stateful IPv4/IPv6 translator [18][19]. NAT-PT nodes are at the boundary between IPv6 and IPv4 networks. Each node maintains a pool of globally routable IPv4 addresses, which are dynamically assigned to IPv6 nodes when sessions are initiated across the IPv6/IPv4 boundary. This mechanism allows native IPv6 nodes and applications to communicate with native IPv4 nodes and applications, and vice versa.

The NAT-PT translation architecture, depicted in Figure 7, also include one or more ALGs (Application Level Gateways). The basic NAT-PT function does not snoop packet payloads, and the application may therefore be unaware of it. Hence, the NAT-PT mechanism depends on ALG agents that allow an IPv6 node to communicate with an IPv4 node and vice versa for specific applications. The NAT-PT mechanism is an interoperability solution that needs no modification or extra software, such as dual stacks, to be installed on any of the end user nodes, either the IPv4 or the IPv6 network. This mechanism implements the required interoperability functions within the core network, making interoperability between nodes easier to manage and faster to manifest

Figure 4: *IPv6 to IPv4* Address Format



3. System Architecture

The main goal of this work is to measure the performance of the tunneling-based mechanisms presented in Section II. The performance of these mechanisms on the real 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.

3.1. Configured-tunnel Testbed

All tunneling mechanisms require that the endpoints of the tunnel run both IPv4 and IPv6 protocols. That is, they must run in a dual-stack mode. Thus, the node which run both IPv4 and IPv6 protocols simultaneously can interoperate directly with both IPv4 and IPv6 end systems and routers. Figure 8 shows a configured-tunnel testbed.

The configured-tunnel mechanism depends on the manual configuration at both end-points: one at the client site and the other at the remote tunnel provider. Once a tunnel has been established, the service provider will advertise the relevant routing information to the client's network. Hence, the end node can support a native IPv6 protocol stack while the edge router generates the tunnel and handles the encapsulation and de-capsulation of IPv6 packets over the existing IPv4 infrastructure. Figure 9 presents the interfaces of the dual-stack gateway.

Figure 5: IPv6 Tunnel Broker

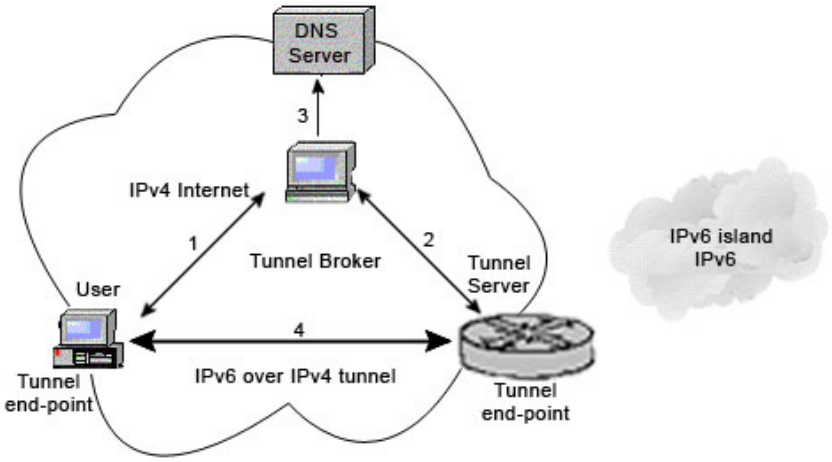


Figure 6: Bump-In-the-Stack Architecture

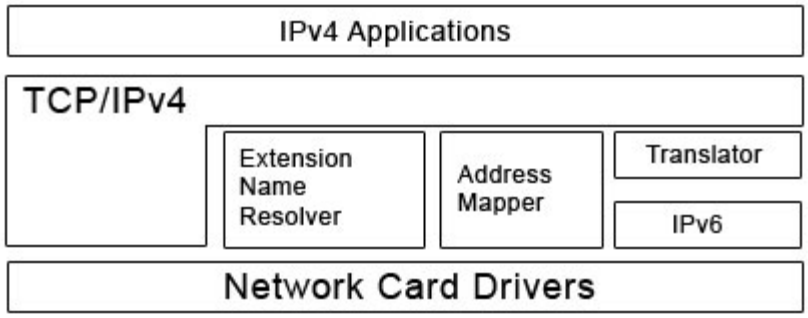


Table 1: Comparisons of Tunnel-based Mechanisms

Tunneling Mechanism	Advantages	Limitations	Requirements
Configured-tunnel Mechanism	Stable and secure links for regular Communication.	Management Overhead; No indepenently managed NAT	ISP-registered IPv6 address Dual-stack router.
6to4-tunnel Mechanism	Connection of multiple remote IPv6 domains.	Number of tunnels supported by the 6to4 router.	IPv6 prefix (2002::/16); Dual-stack router.
Tunnel Broker Mechanism	Standalone isolated IPv6 end systems.	Ptential security implications.	It must know how to create and set a script.

3.2. IPv6 to IPv4-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 (2002:V4ADDR::/48). It can connect to another, 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 network 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. As shown in Figure 10, 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 de-capsulates 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.

The relay router advertises a route to the 2002::/16 network as traffic flows in the reverse direction. 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 de-capsulates the IPv6 packet and forwards it to the IPv6 site. Figure 11 presents the routing table of IPv6 to IPv4gateway.

Figure7: Basic NAT-PT Translation Architecture

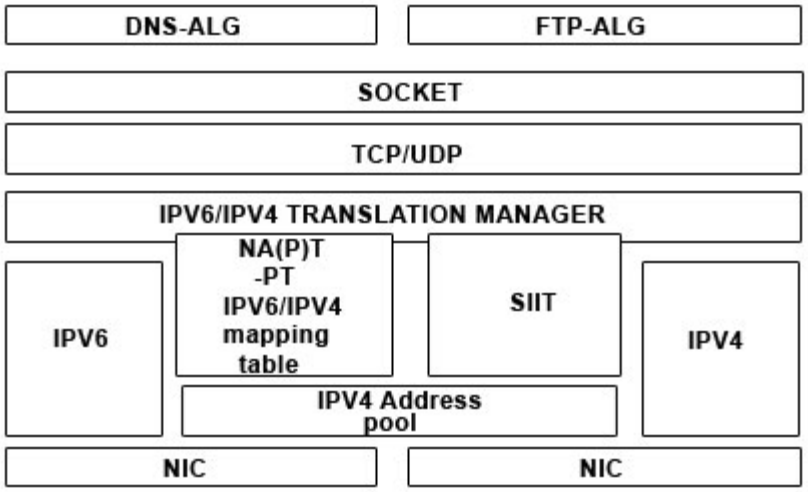
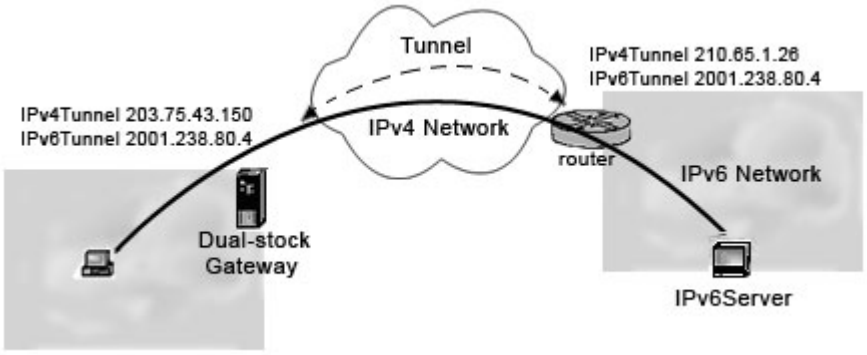


Figure 8: Configured-tunnel Testbed



3.3. Tunnel Broker Testbed

The tunnel broker mechanism requires a dual stack node at the client's end to be connected to the tunnel broker server. The tunnel broker system has a user-friendly interface that enables an isolated IPv6/IPv4 node interactively to establish an IPv6-in-IPv4 tunnel to an IPv6-only network. It is associated with a tunnel server, and is connected using DNS service.

The tunnel broker reduces the management effort required. These services are generally provided through Web-based applications that allocate IPv6 address prefixes and return suitable tunnel configuration scripts (as indicated in Figures. 12 and 13). For example, an enterprise can register the IPv4 address of a remote end system or a router that use IPv4 facilities, with the service provider, over a dedicated website. The service provider delivers a script that lays a tunnel to the IPv6 network, allocates an IPv6 address to the end system, and assigns a network prefix to the router to establish the connectivity of the rest of the site. The tunnel broker manages the generation and deletion of the tunnels, and builds up the Accessibility between dual-stack end systems or IPv6 end systems which connected to dual-stack routers and IPv6 backbone. Table 1 summarizes the advantages, limitations and requirements of tunnel-based mechanisms.

Figure 9 (a): Interfaces of the Dual-stack Gateway IPv6 uninstalled

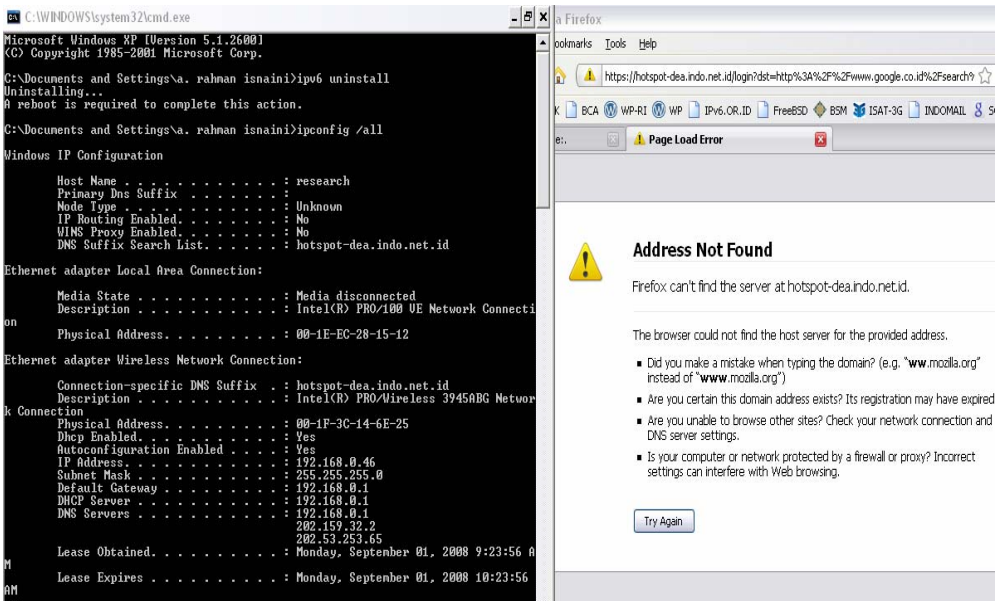


Figure 9 (b): Interfaces of the Dual-stack Gateway IPv6 installed

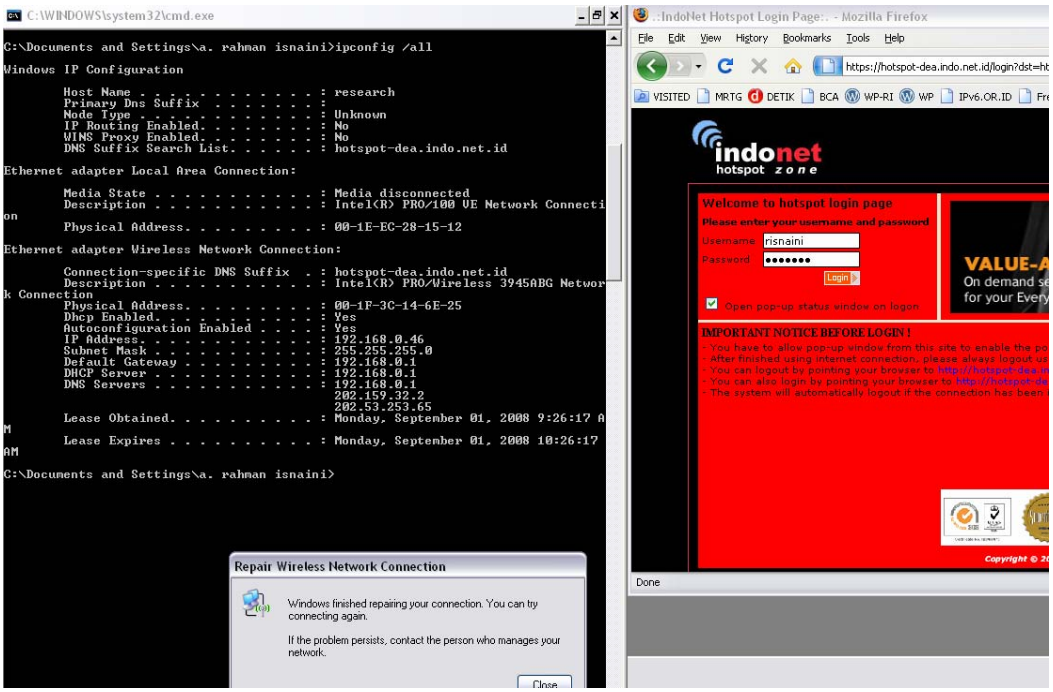
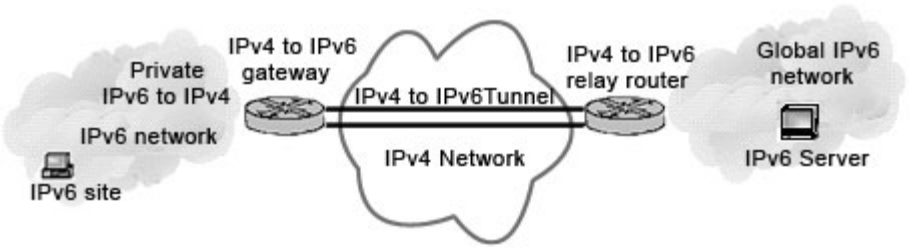


Figure 10: IPv6 to IPv4-tunnel Testbed



4. Performance Analyses

This session describes the performance evaluations of the three tunneling mechanisms in a real network, and presents relevant metrics. Performance analysis is an important reference for designing a good network.

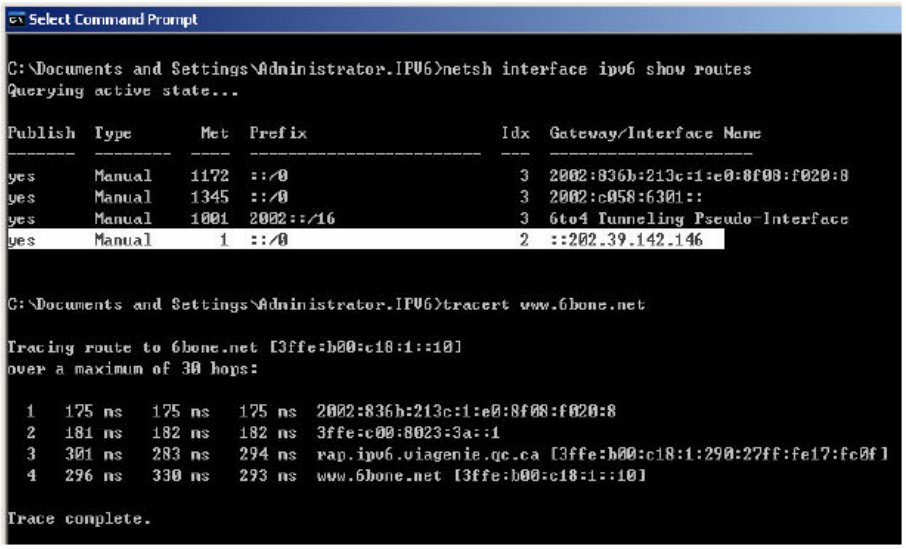
The metrics we used here are latency, throughput, CPU utilization and loss rate.

4.1. Latency Analysis

In evaluating the performance of the tunneling-based 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 ping6 program run on a reliable ICMPv6 Internet layer. The ping6 utility works like its IPv4 counterpart does. It sends ICMPv6 packets to the command argument specified network node and checks the replied message. To determine whether a particular node is alive. Latency was measured by sending different size packets, 64, 128, 256, 512, 768 and 1024 bytes, from a client to a server. 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 more precise result.

Figure 14 shows the comparative latency of the testbed. It presents latency measured as the packet size was varied from 64 bytes to 1024 bytes. It indicates clearly that the 6to4 tunnel mechanism has the least latency, and that the tunnel broker mechanism has the most latency.

Figure 11: Routing Table of the 6to4 Gateway



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. Figure 16 plots the throughput associated with the three mechanisms, for packet sizes that range from 128 bytes to 1024 bytes. The tests were limited to datagram's of 1440 bytes to prevent of a potential undocumented fragmentation problem in the IPv6 protocol stack. In IPv6 devices, the sending node determines the smallest MTU (Maximum Transmission Unit) of the path and fragments the packets accordingly. Hence, fragmentation and reassembling occur only at the source and destination, respectively. IPv6 extension headers contain fragmentation information, and are unaffected by the routers on the path. The maximum throughput is reached for the largest packet sizes, peaking at 69.13 Kbps. The throughput generally increases with the size of the packets. However, the results show that the 6to4 mechanism exhibits the best throughput performance, and the tunnel broker mechanism performs worst. Figure 17 shows the measured results and demonstrates that 6to4 tunnel exhibits the best throughput performance, peaking at around 88.56 Kbps for the largest packets. Figures 16 and 17 show that the configured tunnel and tunnel Broker mechanisms perform very similarly in terms of throughput.

Figure 12: Tunnel Broker Testbed

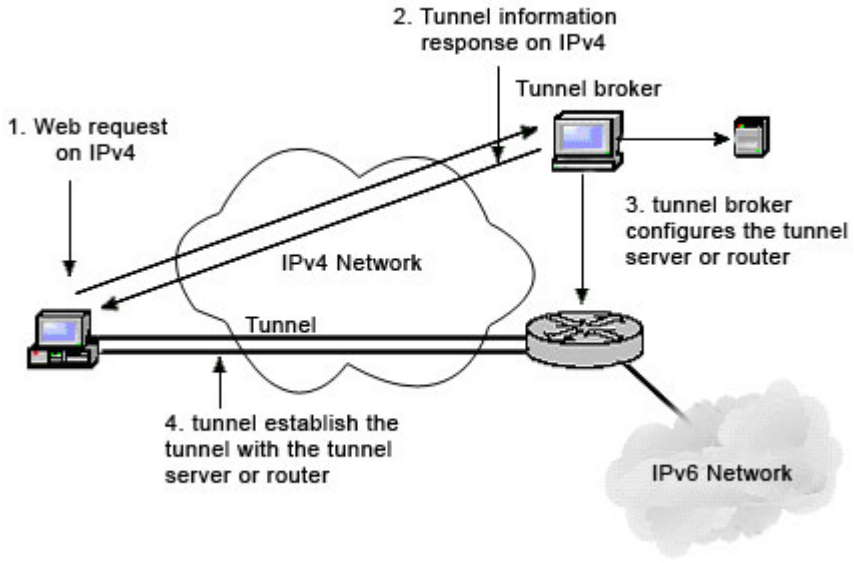


Figure 13: Service Provider Delivers a Script

```
Your request for a tunnel has been accepted

To configure you machine 203.75.43.148, you will need to run
the attached configuration script to start using your tunnel.
As with all scripts you should review its contents to make sure that it
behaves as you expect. You should consult your vendor's documentation to
determine how to make this configuration stable across a machine reboot

attached configuration
ipv6.exe rtu ::/0 2/::203.74.21.3
ipv6.exe adu 2/2001:238:888::1
```

Table 2: Performance Indices for 6bone Site (1: best)

6bone Site	6to4 Tunnel		Configured Tunnel		Tunnel Broker	
	Small packet size	Large packet size	Small packet size	Large packet size	Small packet size	Large packet size
Latency	1	1	2	2	3	3
Throughput	1	1	3	2	2	3
CPU Utilization	3	3	2	2	1	1
Loss Rate	1	1	2	2	3	3

Table 3: Performance Indices for kame Site (1: best)

Kame Site	6to4 Tunnel		Configured Tunnel		Tunnel Broker	
	Small packet size	Large packet size	Small packet size	Large packet size	Small packet size	Large packet size
Latency	1	1	2	2	3	3
Throughput	1	1	2	2	3	3
CPU Utilization	3	3	2	2	1	1
Loss Rate	1	1	2	2	3	3

4.3. CPU Utilization

CPU utilization normally refers to the percentage of CPU time taken by a running process. CPU utilization at the sending node (edge router) was measured using the Windows 2003 Server Task Manager's performance monitoring tool.

Figure 14: Latency Analysis (to the 6bone Site)

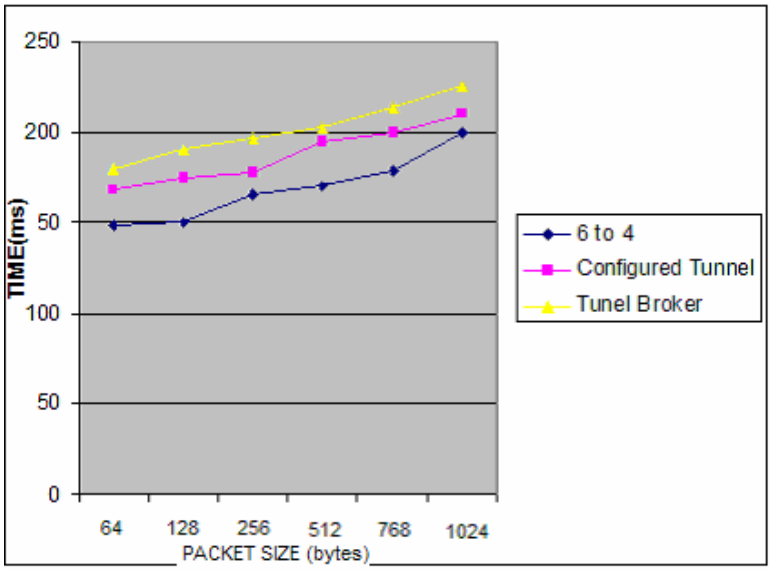


Figure 15: Latency Analysis(to the KAME Site)

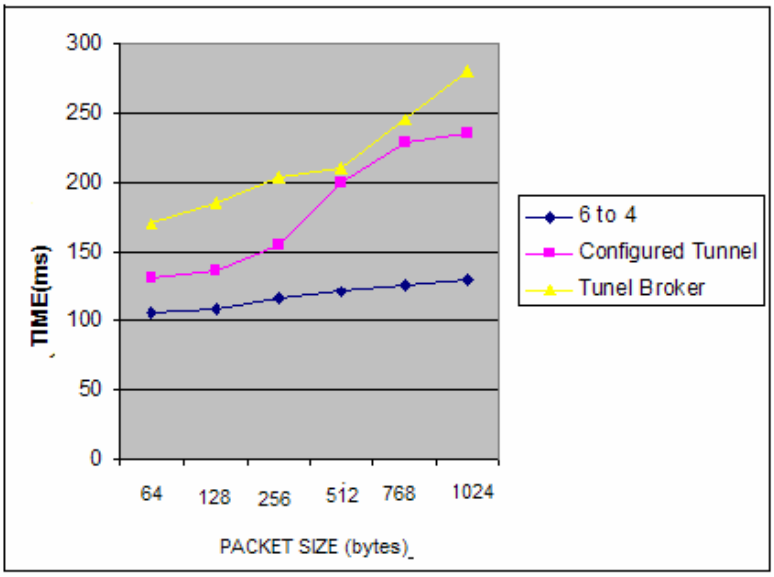
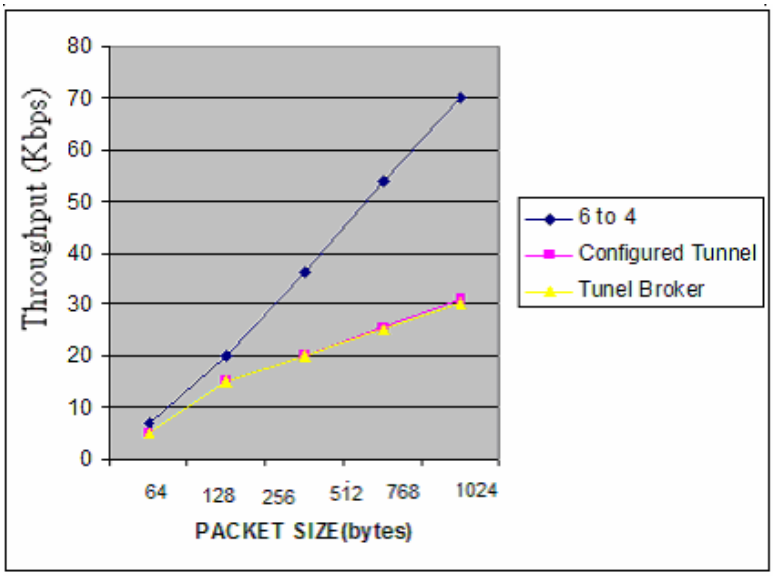


Figure 16: Throughput Analysis(to the 6bone Site)



The 6to4 mechanism makes the greatest CPU utilization because the network node had to do much more work for every packet sent and received, than under the other two mechanisms. A higher CPU utilization of a process corresponds to a higher load on the system; hence, 6to4 mechanism was the most efficient.

4.4. Loss Rate Analysis

In the loss rate analysis, the packet size was increased to measure the corresponding change in the loss rate. Some packets are successfully sent from the client to the server via several network nodes or routers, and some packets are lost unexpectedly reasons.

In Figure 18, when the packet size is 64 bytes, the loss rates of the 6to4 mechanism, configured tunnel and tunnel broker are 1.0%, 1.5% and 1.6%, respectively. When the size of the packet is increased to 1024 bytes, these loss rates become 4.2%, 5.8% and 6.8%. Hence, increasing the packet size increases the loss rate.

Figure 17: Throughput Analyses (to the KAME Site)

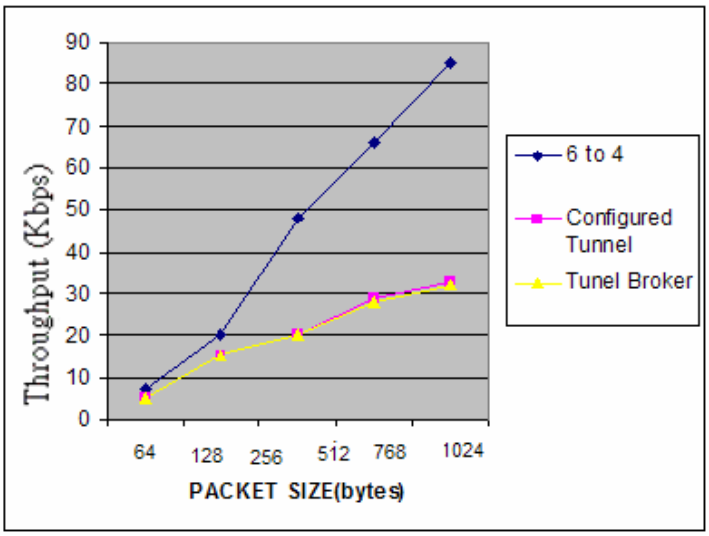
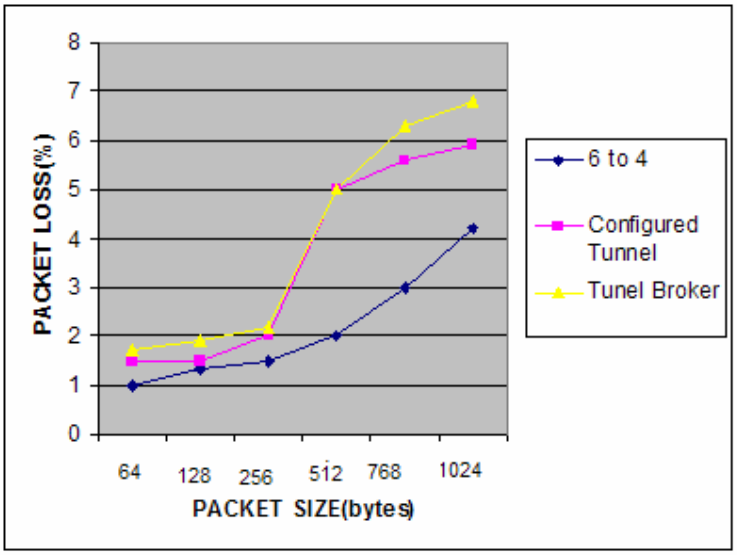


Figure 18: Loss Rate Analyses (to the 6bone Site)



On the kame site, for a packet size of 64 bytes, the loss rates of the 6to4 mechanism, configured tunnel and tunnel broker are 0.2%, 0.8% and 1.2%, respectively. When the packet size is increased to 1024 bytes, the loss rates increase to 1.6%, 2.0% and 2.6%. Tables 2 and 3 summarized the performance metrics.

5. Conclusions

This work examined and evaluated the performance of 6to4 tunnel, configured tunnel and tunnel broker mechanisms in a real network. Tests were performed using a ping6 program run on reliable ICMPv6 packets for connection to a remote IPv6 network such as the 6bone and KAME sites. Different mechanisms have different advantages and disadvantages and may be appropriate for different transition scenarios. The main results of this work are presented below.

- The 6to4-tunnel mechanism will be the solution of first choice. Address allocation is simple and only one tunnel endpoint must to be configured. The 6to4 mechanism forms dynamic stateless tunnels over the IPv4 infrastructure.
- The configured-tunnel mechanism is used to connect IPv6 nodes in the IPv4 Ocean. The tunnel endpoints must be manually configured in the routing table entry. The configured-tunnel mechanism has more feasible because the usage of this mechanism is more strictly controlled to provide greater network QoS, multicast and anycast.
- The Tunnel broker mechanism acts as a virtual IPv6 ISP by providing connectivity among individual sites via IPv6 over IPv4 tunnel. The tunnel broker mechanism depends on a dual stack node at the client's end to be connected to the tunnel broker's facilities. These services are normally provided via Web-based applications that allocate IPv6 address prefixes and return the appropriate tunnel configuration script.

References

[1] T. Dunn, “The IPv6 Transition,” IEEE Internet Computing, Vol.6, No.3, May/June 2002, pp.11-13.
[2] H. Esaki, A. Kato and J. Murai, “R&D Activities and Testbed Operation in WIDE Project,” Proceedings of 2003 Symposium on Applications and the Internet, January 2003, pp.172-177.
[3] IPv6 Forum, The New Internet: Internet for Everyone. (www.ipv6forum.com)
[4] Raicu and S. Zeadally, “Impact of IPv6 on End-user Applications,” Proceedings of the 10th International Conference on Telecommunications, Vol.2, February 2003, pp.973-980.
[5] Microsoft, “IPv6/IPv4 Coexistence and Migration,” White Paper, Washington, November 2001.
[6] Nokia, “IPv6-Enabling the Mobile Internet,” White Paper 10878, Finland, 2000.
[7] J. Davies, Introduction to IP version 6, Microsoft, February 2002.
[8] D. Waddington and F. Chang, “Realizing the Transition to IPv6,” IEEE Communications Magazine, Vol.40, No.6, June 2002, pp.138-147.
[9] S. Hagen, IPv6 Essentials, O'Reilly, July 2002.
[10] K. Wang, A.K. Yeo and A.L. Ananda, “DTTS: a Transparent and Scalable Solution for IPv4 to IPv6 Transition,” Proceedings of the tenth International Conference on Computer Communications and Networks, 2001, pp.248-253.
[11] R. Gilligan, Transition Mechanisms for IPv6 Hosts and Routers, RFC2893, August 2000.
[12] L. Zhou, V. Renesse and M. Marsh, “Implementing IPv6 as a Peer-to-Peer Overlay Network,” Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems, 2002, pp.347-351.
[13] B. Carpenter and C. Jung, Transmission of IPv6 over IPv4 Domains without Explicit Tunnels, RFC2529, March 1999.
[14] C. Huitema, An Anycast Prefix for 6to4 Relay Routers, RFC3068, June 2001.
[15] Durand, P. Fasano, I. Guardini and D. Lento, IPv6 Tunnel Broker, RFC3053, January 2001.
[16] E. Nordmark, Stateless IP/ICMP Translation Algorithm (SIIT), RFC2765, February 2000.
[17] K. Tsuchiya, H. Higuchi and Y. Atarashi, Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS), RFC2767, February 2000.
[18] G. Tsirtsis and P. Srisuresh, Network Address Translation- Protocol Translation (NAT-PT), RFC2766, February 2000.
[19] G.C. Lee, M.K. Shin, H.J. Kim, Implementing NAT-PT/SIIT, ALGs and Consideration to the Mobility Support in NAT-PT environment, Proceedings of the 6th International Conference on Advanced Communication Technology, 2004, pp.433-439.