Performance Investigation of IPv4/IPv6 Transition Mechanisms

Jiann-Liang Chen, Yao-Chung Chang and Chien-Hsiu Lin
Department of Computer Science and Information Engineering
National Dong Hwa University
Hualien, Taiwan
E-Mail: Lchen@mail.ndhu.edu.tw

***Abstract* — 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 transition 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 6to4 mechanism are better than those of the configured tunnel and tunnel broker mechanisms by 89.38% and 94.83%, 42.47% and 48.76%. However, the 6to4 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.**

***Keywords* — IPv6 networks, transition mechanisms, IETF NGtrans, tunneling mechanisms, performance metrics.**

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, opening up a broad field of technological development [3,4]. Companies such as *Microsoft* and *Nokia* have issued white papers on accelerating the IPv6 process [5,6]. Many new applications and Operation Systems, including Windows XP and Linux Kernel 2.1.8 and over, already integrate IPv6 support, but some major challenges remain before an effective and smooth transition from IPv4 networks can be ensured [7].

IPv6-based networks have been implemented in isolation, but now industry is seeking to connect these IPv6 islands over the IPv4 ocean. Much of this work involves a return to simplicity and ease of use with as little disruption the existing networks as possible. Three main transition mechanisms have already emerged dual stack, tunneling and translation, as proposed by NGtrans [8]. 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 disussed.

The next section will introduce various transition mechanisms. Section 3 addresses empirical investigation in this testbed. Section 4 considers performance in terms of transmission latency, throughput, CPU utilization and packet loss. Section 5 discusses results obtained by simulating the system. Finally, Section 6 summarizes this work.

2. IPv4/IPv6 Transition Mechanisms

The transition between today's IPv4 Internet and the IPv6-based Internet of the future will be a long process, during which both protocols will coexist. Figure 1 shows the transition phases. A mechanism for ensuring smooth, stepwise and independent changeover to IPv6 services is required. Such a mechanism must facilitate 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 dual stack, tunneling and translation mechanisms [9]. These mechanisms are briefly described below following.

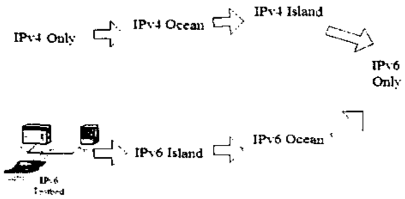

Figure 1. IPv4/IPv6 Transition Phases

2.1 IPv4/IPv6 Dual-Stack Mechanism

Dual stack mechanisms literally include two protocol stacks that operate in parallel and thus allow network nodes to operate via either protocol IPv4 or IPv6 [10]. They can be implemented in both end systems and network nodes. In an system, they enable both IPv4 and IPv6 applications to operate at a single node. Dual-stacked capabilities of network nodes support the handling of both IPv4 and IPv6 packets.

In a dual-stack mechanism, specified in IETF RFC2893, a network node includes both IPv4 and IPv6 protocol stacks in parallel (Fig. 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 transitioning solution. However, dual stack mechanisms enable only similar network nodes to communicate (IPv6-IPv6 and IPv4-IPv4). Much more work is required to create a complete solution that supports IPv6-IPv4 and IPv4-IPv6 communications.

IPv4 applications	IPv6 applications
Sockets API	
UDP/TCPv4	UDP/TCPv6
IPv4	IPv6
L2	
L1	

Figure 2. Dual-stacks Transition Mechanism

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- *6over4*, *6to4* automatic tunneling, and Tunnel Broker.

A. 6over4 Mechanism

The *6over4* mechanism embeds an *IPv4* address in an *IPv6* address link layer identifier part, as shown in Fig. 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 *6over4* tunneling setup provides a solution for IPv6 nodes that are scattered throughout the base 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.

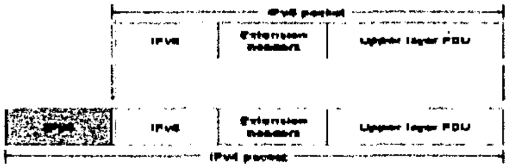

Figure 3. *6over4* Address Link Layer Identifier

B. 6to4 Automatic Tunneling Mechanism

Automatic tunneling refers to a tunnel configuration that does not need direct management. An automatic *6to4* 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 *6to4* 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.

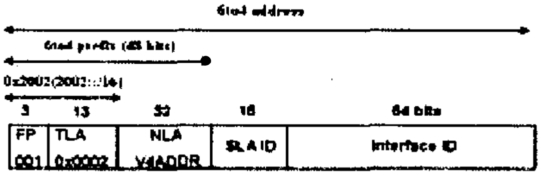

Figure 4. *6to4* Address Format

C. 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 (Fig. 5).

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

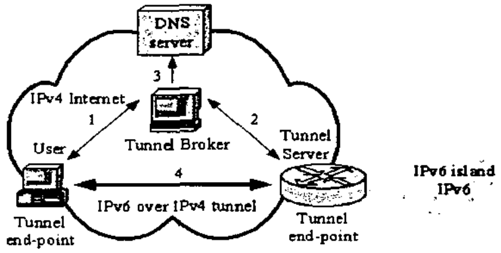

Figure 5. IPv6 Tunnel Broker

2.3 IPv4/IPv6 Translation Mechanisms

The basic function of translation in IPv4/IPv6 transitioning 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, which are described below.

A. 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 (Fig. 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.

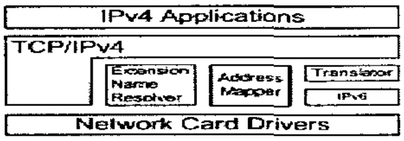

Figure 6. Bump-In-the-Stack Architecture

B. Network Address Translation-Protocol Translation

The NAT-PT mechanism is a stateful IPv4/IPv6 translator [18]. 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 arrangement allows native IPv6 nodes and applications to communicate with native IPv4 nodes and applications, and vice versa.

The NAT-PT translation architecture, depicted in Fig. 7, also include an ALG (Application Level Gateways). The NAT-PT mechanism does not snoop 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. 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.

3. System Architecture

The main aim of this work is to measure the performance of the tunneling-based mechanisms presented in Section 2. The effects of these mechanisms on the performance of a real network, including nodes and routers that support dual IPv4/IPv6 stacks, were examined. Tunneling supports early IPv6 implementation and the use of the existing IPv4 infrastructure without changing the IPv4 modules. The following sections describe three tunneling architectures used for testing herein.

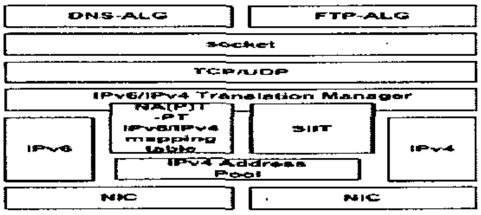
DNS-ALG
FTP-ALG
socket
TCP/UDP
IPv6/IPv4 Translation Manager
NA(P)T-PT IPv6/IPv4 mapping table
SIIT
IPv6
IPv4
IPv4 Address Pool
NIC
NIC

Figure 7. Basic NAT-PT Translation Architecture

3.1 Configured-tunnel Testbed

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

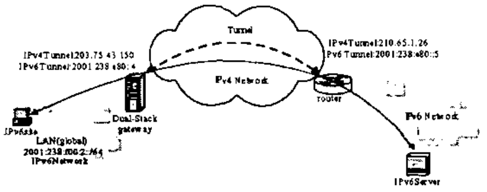

Figure 8. Configured-tunnel Testbed

The configured-tunnel mechanism depends on the configuration of 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 9. Interfaces of the Dual-stack Gateway