CHAPTER 3

IPV4/IPV6 TRANSITION TECHNIQUES

This chapter demonstrates the requirement for a transition technique between IPv4 and IPv6 in section 3.1. Several transition techniques have been created to fill this need. They have their own properties. An overview of each technique is presented in section 3.2. The selected technique and the relevant works are introduced in section 3.3. The summary is in section 3.4.

3.1 Introduction

IPv6 is the long term solution to solve the address depletion problem of IPv4. The initial plan was to install IPv6 in all nodes for the existing system. This technique is called Dual-Stack [7]. Nodes handle both IPv6 and IPv4. A node which wants to connect to the IPv6 domain uses IPv6 while it still works when connecting to the IPv4 domain by using IPv4. This technique needs an IPv4 address to be available for each Dual-Stack node. Eventually, when there are no IPv4 addresses left, a new node connecting to the network must be v6 only..

In the early stage of migration, several networks deploy IPv6. At that time, v6 networks look like islands in v4 ocean – there are some v6 networks among a huge of v4 networks. Each v6 network wants to connect to each other. The communication between them can be accomplished by using tunneling techniques which provide the communication between IPv6 nodes over the IPv4 network where native IPv6 connectivity is not available. Tunneling consists of the encapsulation and decapsulation of packets. The nodes send encapsulated IPv6 in IPv4 packets over the IPv4 network.

Some of the existing networks still use IPv4 only and will continue as long as IPv4 work. They do not want to change to the new IP. When some v6 only networks are attached in the Internet, it is possible to have connection from a v6 node in these v6 sites to the existing network. However, the v6 node cannot connect to the existing world because of the different protocol. A translator is required to provide communication between them.

The translation to IPv6 encounters to several obstacles. For example, IPv4 and IPv6 are incompatible and cannot communicate to each other. The large number of computers and network devices are in the Internet. They need cost and time to upgrade their system to support IPv6. Several of the existing applications must be improved to support IPv6. Some do not accept to use the new IP. So several network devices in the old world still use IPv4 to provide their communications.

In order to allow the communication between IPv4 and IPv6, it needs the method to move the Internet from using IPv4 to IPv6 without requiring all nodes to be changed simultaneously. A method to allow IPv4 and IPv6 needs to communicate with each other is needed as part of the transition plan. This will allow IPv4 only and IPv6 only networks to co-exist.

3.2 IPv4/IPv6 Transition technique

This work focuses on the transition mechanism that provides applications and communications between IPv4 and IPv6. IETF has presented several transition strategies to provide the communication. The following section explains the overview of these transition techniques.

3.2.1 Dual-stack

This technique is in the early transition mechanism which makes a node to become the complete IPv6 and IPv4. The node which wants to connect to IPv6 domain needs to install IPv6 program. And it still works well when connects to IPv4 domain by using IPv4 address. This technique is called Dual-stack [7]. The node which uses this technique is called a Dual-stack node. It has IPv6 and IPv4 protocol which independently work at each layer of OSI model. Both of them are installed in each node. The structure of Dual-Stack is shown in Figure 3.1.

Dual-Stack is installed in a host or router which is usually as a tunnel end-point or an individual host. It can directly communicate with IPv4 and IPv6 hosts. IPv4 address is assigned by manually configured or DHCP server and IPv6 address is also assigned by manually configured or IPv6 stateless/statefull configuration mechanism.

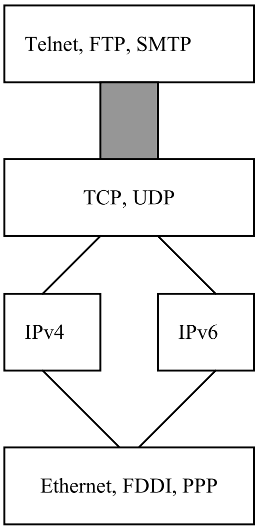

Figure 3.1 Structure of Dual-Stack

As a consequence, the DNS resolver library of he dual-stack host can use either A and AAAA type records. When the host wants to resolve a name or address of destination, it sends A and/or AAAA query to the DNS resolver. If the returned answer is A type, the host uses IPv4 stack. If the returned answer is AAAA type, the host uses IPv6 stack. In the case of A and AAAA are returned, the host can choose either A or AAAA to connect to the destination. It might choose the answer that it gets first.

Advantages

1. This technique provides independently communication to IPv4 and IPv6. A Dual-Stack node can chose either IPv4 or IPv6 address to connect to a destination node. It depends on the destination node whether IPv4 or IPv6.
2. The Dual-Stack node can use both v4 and v6 applications from v4 and v6 networks.

Limitations

1. This technique needs IPv4 address as the important component. If it is widely deployed in a large network, IPv4 address depletion might be a critical problem.
2. It is necessary to install both IPv4 and IPv6 address into every node that wants to use both v4 and v6 applications.

3.2.2 A SOCKS-based IPv6/IPv4 Gateway Mechanism (SOCK64)

This mechanism applies the SOCKS protocol [10] to the heterogeneous mechanism communication and relays two terminated IPv4 and IPv6 connections at the application layer over IPv4 infrastructure. This technique is the SOCKS-base IPv6/IPv4 gateway mechanism and it can provide the same communication environment as native SOCKS mechanism without the modification a new protocol. There are two major components as follows:

SOCv5 library: is installed in each client then the client in this mechanism is called "socksifying host". This library is as a cushion between the application and the transport layers. Then it is not necessary to change anything of the application because a cushion layer translates the sock API as application.

SOCK64 server: is installed on dual-stack node which acts as a gateway for enabling interconnection between the difference IP realm as in Figure 3.1

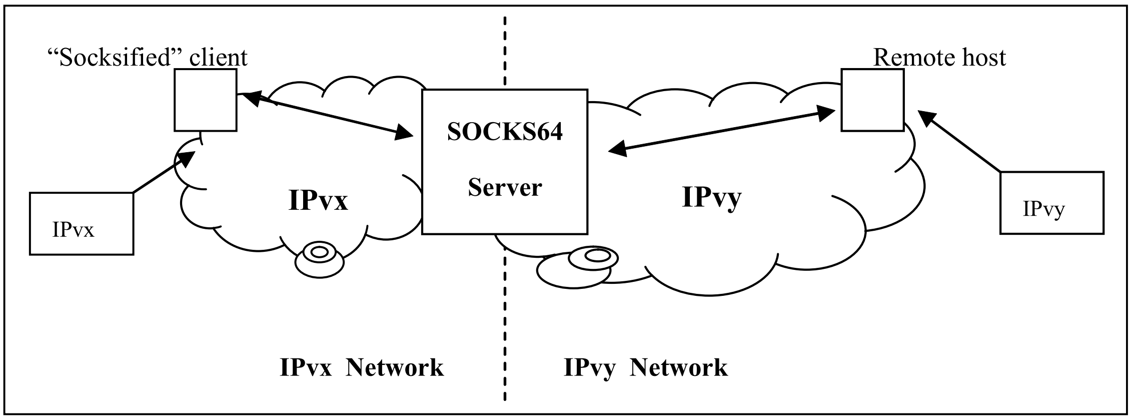

Figure 3.1 Structureof SOCK64

Advantages

1. The command of native SOCKv5 protocol can be use in this mechanism.
2. It is not necessary to modify the existing application to support IPv6 because SOCK64 changes the link order of dynamic link library.
3. A DNS name and address resolution is replaced by a DNS name resolving delegation at SOCK library.

Limitations

1. Only socksified host can initiate the sock-base connection.
2. It does not provide end-to-end security between the original source and the final destination.
3. Its function (e.g., getpeername() and getsockname()) cannot provide valid IP as a return value.

3.2.3 An IPv6-to-IPv4 Transport Reply Translator (TRT)

The motivation of this mechanism is to allow an IPv6-only host inside IPv6 realm to exchange (TCP, UDP) traffic with an IPv4-only host inside IPv4 realm. This technique is called IPv6-to-IPv4 Transport Relay Translator (TRT) [9]. It is placed at the border gateway between IPv6 and IPv4 realms as shown in Figure 3.1. It acts as the relay between different IP realms. One side of TRT is as a pretend destination while another is as a pretend source. In corresponding, one side of TRT which is attached on IPv6 realm is marked as "dummy prefix". It dose not actually exist while another side attached on real IPv4 realm. For TCP, then there are two TCP connections. One is TCP/IPv6 and another is TCP/IPv4.

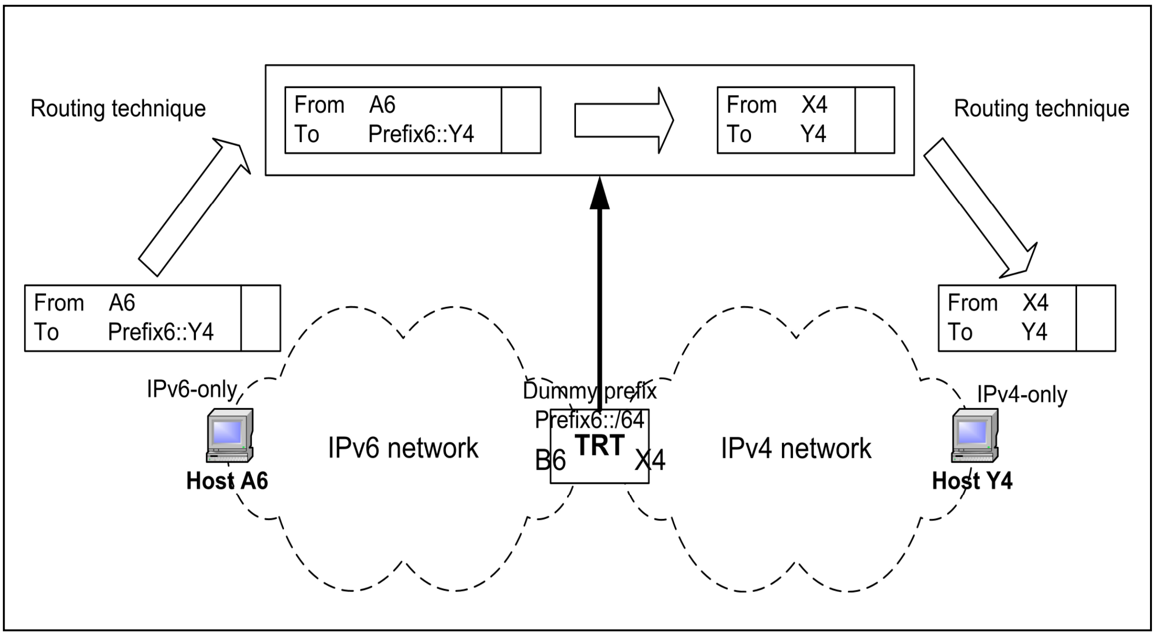

Figure 3.1 The structure of TRT communication

When the initiating host whose IPv6 address is A6 wants to connect to the destination IPv4 host whose IPv4 address is Y4, it needs to make TCP/IPv6 connection to Prefix6::Y4.

The packet is routed to the TRT system. The system accepts the TCP/IPv6 connection between A6 and Prefix6::Y4. And it communicates with the initiating host using TCP/IPv6. Then it checks the last 32 bits if the destination address (the IPv6 address Prefix6::Y4) to the actual IPv4 address (the IPv4 address Y4). The TRT system makes the new TCP/IPv4 connection from X4 to Y4 and sends the traffic across the two TCP connections.

There are two connections for each communication. One is TCP/IPv6 and another is TCP/IPv4: from A6 to B6, and X4 to Y4. TRT is the relay to create a connection between v4 and v6 networks. Then it needs the table to keep a state of address mapping and connection of each v4 and v6 communication. These procedures are also applied to relay UDP.

Advantages

1. It dose not need the extra modification in both IPv6-only source host and IPv4-only destination host.
2. TRT does not maintain part MTU and fragmentation because these procedures are provided by the IPv6-to-IPv4 header converters.
3. IPv6 path MTU is independently decided from IPv4 path MTU.
4. When a special DNS server implementation is combined with TRT, this supports IPv6-to-IPv4 translation very well and TRT can cover most of the daily applications.
5. It is possible to have multiple TRT systems which can be installed very easily.

Limitations

1. It needs statefull TRT system and it can be considered a single point of failure.
2. It cannot translate protocols that are not known to the TRT system.
3. For this technique, IPsec cannot be used across a relay.

3.2.4 Network Address Translation – Protocol Translation (NAT-PT)

The intended of this mechanism is to provide transparent routing an IPv6 end node and applications in v6 realm to communicate with IPv4 end node in other and vice versa. The aim is to translate between IPv6 and IPv4. NAT-PT [8] is as a function of router to provide the translation at the border of network as shown in Figure 3.1. Then the communication can be provided without changing both end nodes. A NAT-PT box has a pool of global unique IPv4 addresses. It is assigned to IPv6-only node when IPv6-only node establishes a session from inside to IPv4-only node outside network. The NAT-PT router keeps the state information of translation until the session is terminated. It has the table to contain address mapping between IPv4 and IPv6. Like NAT, NAT-PT always interacts with Application Level Gateway to provide application.

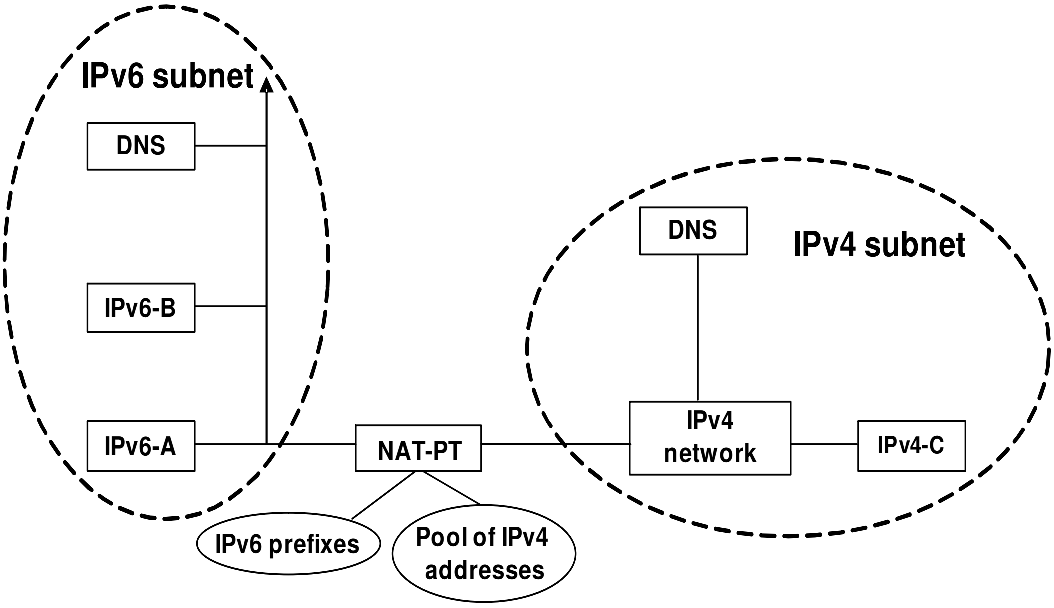

Figure 3.1 Structure of NAT-PT

It has two variations: Traditional-NAT-PT and Bi-directional-NAT-PT as follows:

1. **Traditional NAT-PT**

Traditional NAT-PT allows v6 hosts in a v6 network to access v4 hosts in a v4 network. Like traditional-NAT, there are two types to traditional-NAT-PT, Basic NAT-PT and NAPT-PT, which provide uni-directional outbound from IPv6 to IPv4 domain.

Basic NAT-PT: it performs protocol and address translation from IPv6 to IPv4. The translation impacts to the upper layer protocol such as TCP/UDP and ICMP header checksums. They must be updated following the address changing. NAT-PT keeps the address mapping of the session into the mapping table. The inbound session uses the same address mapping as the outbound.

NAPT-PT: This way is extended to translate transport identifier such as TCP and UDP port numbers and ICMP query identifier. It allows the transport identifier of v6 hosts are multiplexed into the transport identifier of a single IPv4 address. The single address is allowed to share by a set of v6 hosts. NAPT-PT can be combined with Basic-NAT-PT. Then the pool of external addresses is used by Basic NAT-PT and NAPT-PT to cooperate with port translation.

2. Bi-directional NAT-PT

Bi-directional NAT-PT is contrast with traditional NAT-PT which permits a communication from v6 to v4 only. It provides the translation from v4 to v6 and vice versa. A communication can be initiated from v4 hosts as well as v6 hosts.

NAT-PT has two main functions to provide communication between IPv4 and IPv6: protocol and address translation. It also cooperates with the ALG to provide unicast application.

3.2.4.1. Protocol translation

Although NAT-PT was derived from NAT, a different function of them is protocol translation. This work is provided by NAT-PT but it is not available in NAT. NAT performs address translation between private and global IPv4 addresses which are the same protocol.

On the other hand, a communication between IPv4 and IPv6 hosts is a connection between different protocols. It is necessary to translate either IPv4 or IPv6 to the other protocol before the connection is established because both end nodes need to be visible to each other. NAT-PT is based on Stateless IP/ICMP Translation [17] algorithm to process protocol translation. But it must modify something to perform network address translation.

To translate protocol, NAT-PT changes header of an incoming packet from either IPv4 or IPv6 to the other. Not only protocol translation but also IP addresses are required to be translated in order to allow both end nodes communicate to each other under the same rule. Address translation which will be presented in next section is a part of protocol translation. The address must be translated to be the appropriate IP version in order to visible a recipient to get a packet from a sender.

The upper layer protocol, TCP/UDP, gets the impact of protocol translation. When IP header in the packet is translated, checksum in TCP/UDP header must be recalculated because protocol and IP address are changed. If the packet is defined to translate port number, the port field in TCP/UDP header is replaced with the mapped port which is registered in the mapping rules.

In addition, the other protocols which relevant with IP such as ICMPv4 [14] and ICMPv6 [18] are necessary to be changed following the protocol translation as ICMP has also been updated for IPv6.

3.2.4.2. Address translation

Address translation is operated to change source and destination addresses to be the appropriate numbers in order to provide communication between IPv4 and IPv6 nodes. Source and destination IP addresses of a packet must be known to both end nodes. NAT-PT must translate the addresses from either IPv4 or IPv6 to the other before sending the packet to the destination network. It needs a particular IP address or a block of addresses to process address translation.

IP addresses of these end nodes are not modified on the other hand their transmitted packets are translated by NAT-PT at the border of the network. NAT-PT relies on the mapping rules and uses the particular address as a temporary number to perform address translation.

For an outbound session from an IPv6 to an IPv4 node, the IPv6 packet including its addresses must be translated to IPv4. In this case, NAT-PT requires a temporary IPv4 address to translate the IPv6 address. A particular IPv4 address or a block of addresses can be assigned in an address pool. When NAT-PT gets the incoming v6 packet, it looks for IPv6 address of the packet in the rules. If the address is found, NAT-PT translates the packet and address to IPv4. Likewise, IP packet in reverse path relies on the similar address translation. The translation relies on the same address mapping as long as the communication is connected between the same source and destination nodes.

The other direction, when an IPv4 node wants to connect to an IPv6 node, an IPv4 packet and its addresses must be translated to IPv6. The same procedures as the previous case are performed but NAT-PT requires a temporary IPv6 address to translate the IPv4 packet. NAT-PT uses a particular IPv6 address format as the temporary IPv6 address. It has a simple way to generate the temporary IPv6 address. NAT-PT assigns a particular IPv6 unicast prefix which is attached in the front of IPv4 address of the translated packet. Then the translated address looks like **IPv6-unicast-prefix::IPv4-address/prefix-length**. In reverse path, NAT-PT maps the translated address to IPv4 address by removing the IPv6 prefix. Then the remainder is the original IPv4 address.

3.2.4.3. Address discovery

To communicate in the Internet, like the postal system, a sender needs to know the address of a receiver. It usually looks up a destination address from DNS and then sends packets of data out through a path to the receiver. A communication between v4 and v6 nodes likes a communication between two nodes which have the same IP protocol. A v6 source node wants to know IPv6 address of destination while a v4 source node requires IPv4 destination address. They can discover their destination addresses from DNS server. However, IP address in a DNS reply message must be the appropriate address for each source node in order to allow a connection between nodes which have difference IP protocols.

The original IP address in a DNS answer cannot be used as a destination number for a source which has different IP version. Like NAT, NAT-PT cannot translate IP address inside payload of DNS packet. It needs a helper to adjust the destination address to be the adequate IP address which is Domain Name System-Application Layer Gateway (DNS-ALG) [13] as shown in Figure 3.1.

DNS-ALG can provide two ways address mapping from IPv6 to IPv4 and vice versa. It modifies an address in DNS message to be an appropriate IP version. Then it tells to NAT-PT about mapping of the original and the temporary addresses. The mapping is recorded into the list as the rules for translation.

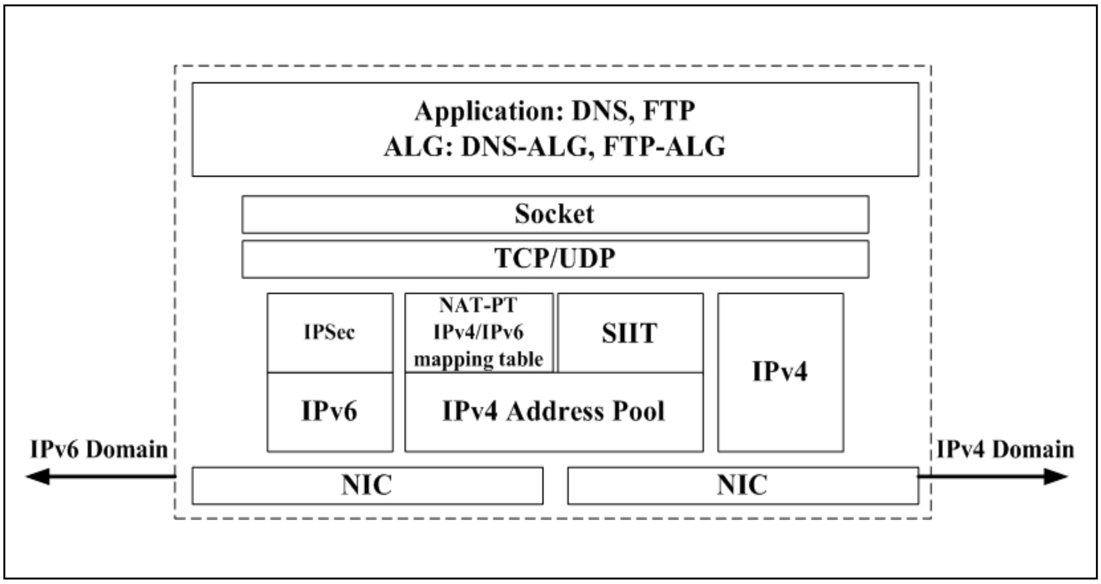

Figure 3.1 DNS-ALG communication

Operation of DNS-ALG to perform address mapping from IPv4 to IPv6

When a v6 node wants to connect to a v4 node, it wants to know IPv6 address of the destination. It looks up destination address from DNS server by sending a v6 DNS query. DNS-ALG who listens the v6 query generates the other query that find the other IP version of the same destination node. It sends its own the v4 query to DNS server. Figure 3.2 shows diagram of DNS-

ALG to perform address mapping from IPv4 to IPv6.

Since the destination is IPv4 node, only IPv4 address or A record is available in DNS server. When the reply packet which contains IPv4 address only is sent back to the source, DNS-ALG modifies the IPv4 address to be the IPv6 address. To process this work, the ALG allocates a temporary IPv6 address as the destination number and send the answer to the v6 source.

The simplest way to generate the temporary IPv6 is to assign a particular IPv6 unicast prefix with length 96 bits for DNS-ALG. It is combined in a front of the IPv4 address in DNS reply message. The v6 source uses the translated IPv6 address as the destination number.

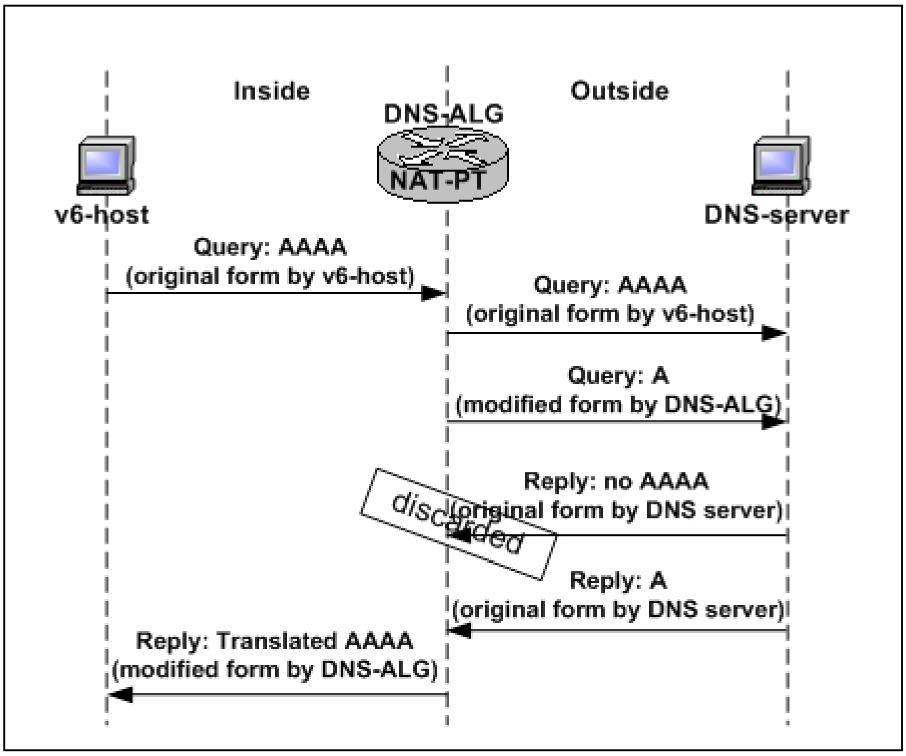
Inside
DNS-ALG
Outside
NAT-PT
v6-host
DNS-server
Query: AAAA
(original form by v6-host)
Query: AAAA
(original form by v6-host)
Query: A
(modified form by DNS-ALG)
Reply: no AAAA
(original form by DNS server)
discarded
Reply: A
(original form by DNS server)
Reply: Translated AAAA
(modified form by DNS-ALG)

Figure 3.2 DNS-ALG address mapping for IPv6 to IPv4

As the migration to IPv6 proceeds, the IPv4 node is likely to be assigned an IPv6 address to provide the existing applications for nodes in v6 networks. This IPv4 node becomes Dual-Stack. Both IPv4 and IPv6 addresses of that node are registered in DNS server. Then there are two DNS reply messages that contain A and AAAA records as the address type of a destination node. This scenario is shown in Figure 3.3.

In this case, DNS-ALG does not perform address mapping. The DNS reply packet with the authenticated IPv6 address should be sent back to the IPv6 source. The other packet should be dropped.

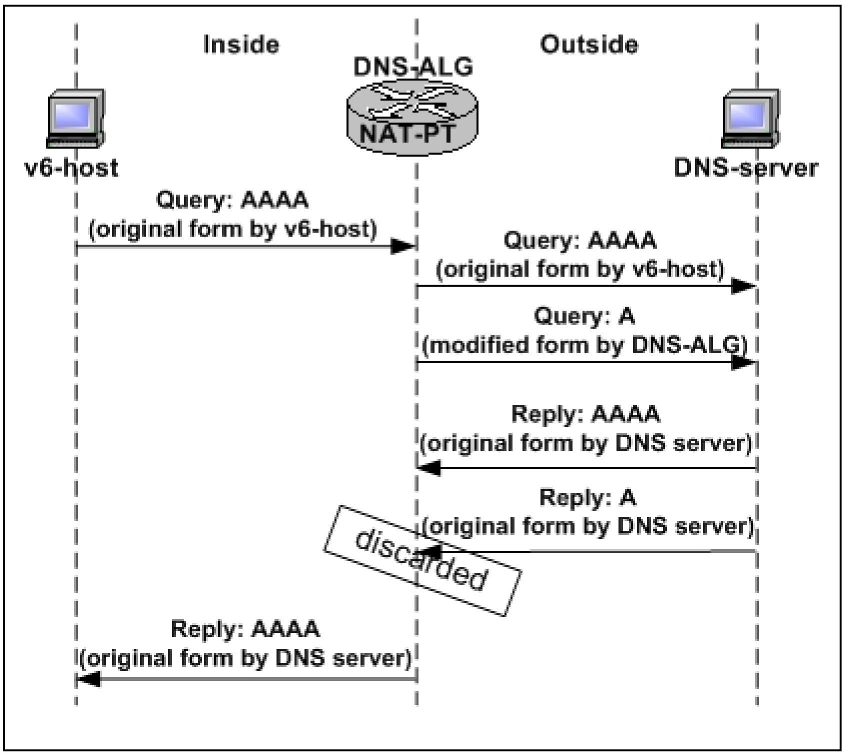
Inside
DNS-ALG
Outside
NAT-PT
v6-host
DNS-server
Query: AAAA
(original form by v6-host)
Query: AAAA
(original form by v6-host)
Query: A
(modified form by DNS-ALG)
Reply: AAAA
(original form by DNS server)
Reply: A
(original form by DNS server)
discarded
Reply: AAAA
(original form by DNS server)

Figure 3.3 DNS-ALG address mapping for Dual-Stack on destination v4 node

Operation of DNS-ALG to perform address mapping from IPv4 to IPv6

An IPv4 source node looks up IPv4 destination address from DNS server when it wants to connect to a destination node. This case is assumed that the destination node is configured with IPv6 only. In DNS server, its address is registered with AAAA address type only.

This connection is processed like the procedures in the previous case, from IPv6 to IPv4. DNS-ALG intercepts and modifies the answer packet of DNS which contains IPv6 address as the destination number. In the DNS packet, the ALG replaces the existing IPv6 address with a temporary IPv4 address. The temporary address is allocated from a particular IPv4 or a block of IPv4 addresses which is assigned for NAT-PT translation. The IPv4 source uses the translated address in the DNS packet as the destination. Figure 3.8 shows diagram of DNS-ALG to provide address mapping from IPv4 to IPv6.

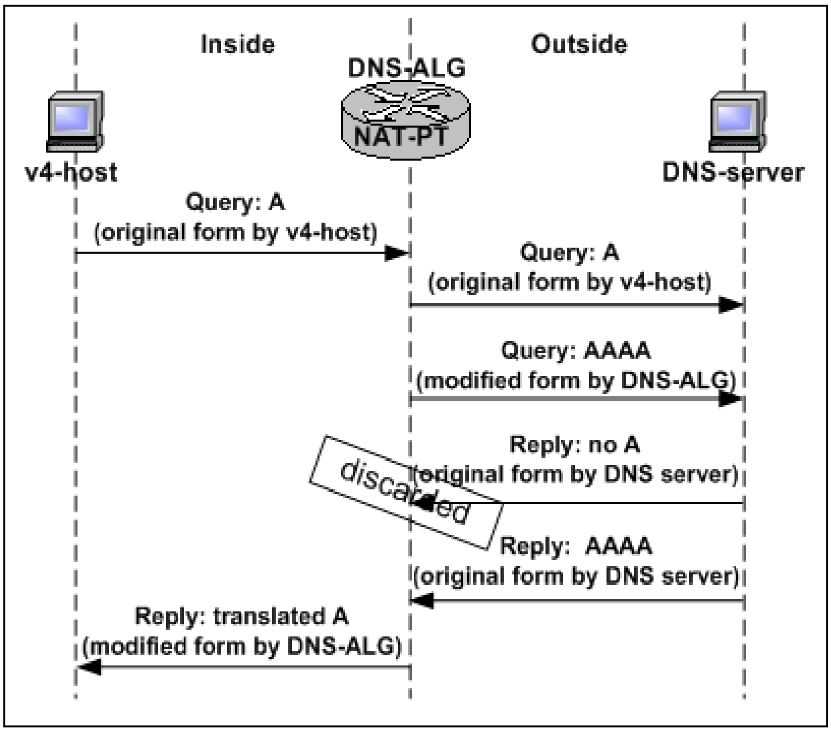
Inside
DNS-ALG
Outside
NAT-PT
v4-host
DNS-server
Query: A
(original form by v4-host)
Query: A
(original form by v4-host)
Query: AAAA
(modified form by DNS-ALG)
Reply: no A
(original form by DNS server)
discarded
Reply: AAAA
(original form by DNS server)
Reply: translated A
(modified form by DNS-ALG)

Figure 3.4 DNS-ALG address mapping for IPv4 to IPv6

However, it is possible that a destination is Dual-Stack node and it has both IPv4 and IPv6 addresses in DNS server. Then there are two DNS reply messages that contain A and AAAA records as the address type of a destination node as shown in Figure 3.5. In this case, DNS-ALG does not perform address mapping. The DNS reply packet with the authenticated IPv4 address is sent to the IPv6 source while the other is dropped.

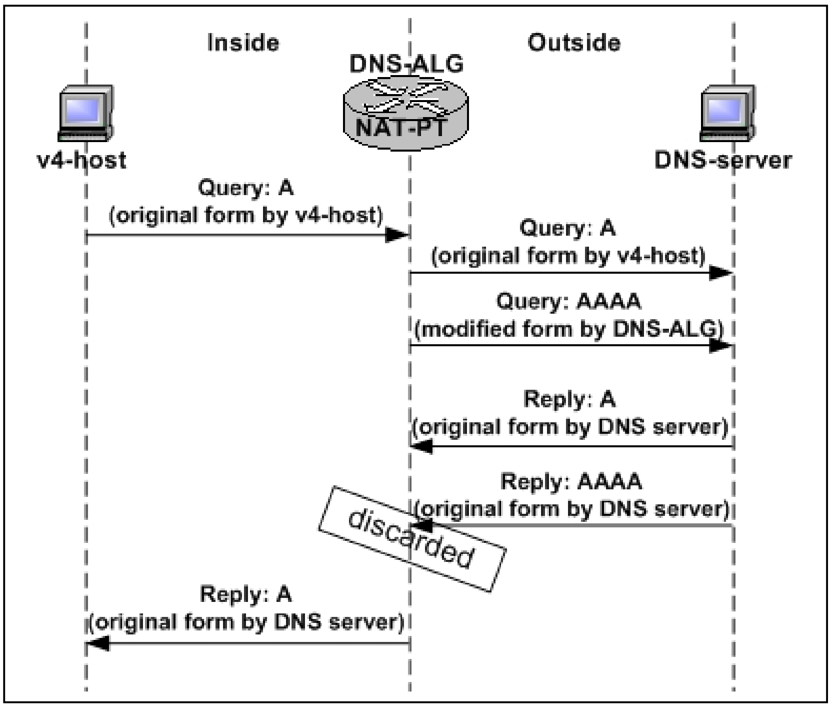
Inside
Outside
DNS-ALG
NAT-PT
v4-host
DNS-server
Query: A
(original form by v4-host)
Query: A
(original form by v4-host)
Query: AAAA
(modified form by DNS-ALG)
Reply: A
(original form by DNS server)
Reply: AAAA
(original form by DNS server)
discarded
Reply: A
(original form by DNS server)

Figure 3.5 DNS-ALG address mapping for Dual-Stack on destination v6 node

By using the cooperation between DNS-ALG and NAT-PT as above, IPv4 and IPv6 source nodes are discovered from several scenarios as the summary in Table 3.1.

Table 3.1 Address discovery of IPv4 and IPv6 source nodes

Node A Node B	IPv4-only	Dual-stack	IPv6-only
IPv4-only	IPv4 address	IPv4 address	Translated(IPv6) to IPv4 address
Dual-stack	IPv4 address	IPv4 or IPv6 address	IPv6 address
IPv6-only	Translated(IPv6) to IPv4 address	IPv6 address	IPv6 address

Advantages

1. NAT-PT technique can develop to be a tool or function of border router between v6 and v4 realm.
2. The useful of uni-directional type NAT-PT without DNS-ALG, is visible v6 server inside v6 network from external network while the outbound sessions go on.
3. The useful of bi-direction type is to visible internal and external network of NAT-PT.

Limitations

1. This mechanism does not provide end-to-end security same as NAT.
2. It must interact with an ALG to support the applications that have IP addresses embedded in their payloads. For critical applications, NAT-PT may configure static address mapping between v6 and v4 address.
3. Dynamic address that is reused by NAT-PT might not be the same IP for applications.
4. It needs to via a single point to process address translation. A connection that has address mapping in NAT table breaks when a NAT-PT router fails.

The mechanisms as described above can be summarized as in Table 3.2. Dual-Stack needs IPv4 and IPv6 address as a component. It can provide completely communication to v4 and v6 networks. It is very close to a critical problem of IPv4 address depletion if every node and router are deployed Dual-Stack. SOCK64 is a translator at application level to allow a v4 node to use v6 applications from a v6 site and vice versa. And it requires SOCK64 server to be a gateway to enable interconnection between v4 and v6 networks. End nodes must be modified or installed the particular application if it is required to deploy these techniques, Dual-Stack or SOCK64.

TRT is a relay translator at transport level to create TCP/UDP connection between IPv4 and IPv6. It can provide a communication between v4 and v6 nodes without changing at the end. But it stays on statefull to keep a status of TCP/UDP connection and address mapping between IPv4 and IPv6. NAT-PT is a translator at network level to translate address and packet between

IPv4 and IPv6. Then it provides transparency communication between IPv4 and IPv6 nodes. And it is similar to NAT which has already been used.

Table 3.2 The comparison of IPv4/IPv6 transition techniques

Transition mechanism	**Apply**		**Connection**		**Concept**
	Router	**End node**	**4-to-6**	**6-to-4**	
Dual-Stack	Yes	Yes	Yes	Yes	Independent IPv4 and IPv6 communication
SOCK64	Yes	Yes	Yes	Yes	Socket application, modify end node, required server
TRT	Yes	No	Yes	Yes	Translate transport TCP/UDP, stateful
NAT-PT	Yes	No	Yes	Yes	Translate header packet

3.3 The altered technique and literature review

Since NAT-PT was derived from NAT, they have the same features in some functions. NAT-PT program is available to implement for testing and it is expected to be used in the migration period. From the surveyed transition techniques in section 3.2, NAT-PT has several advantages. It can provide bidirectional unicast communications and applications between v4 and v6 worlds. NAT-PT translations allow transparency communication between the protocols without changing anything at the end nodes. The outstanding is a motivation to evaluate and enhance NAT-PT's features.

Before moving to the section to investigate NAT-PT, the works which relevant to study and develop NAT-PT are presented.

After NAT-PT was submitted to be the IETF Internet standard by Tsirtsis and Srisuresh, it has been tested and studied several times. Fiuczynski, et al., [19] presented the design and the implementation of the address and protocol translation. They found that this technique can provide address and protocol translation and support applications e.g. HTTP, FTP, SMTP. However, there are several fields cannot be translated from one to the other, the impact of translation is lost the information in the translated packet.

Jacobsson and Sorani [20] presented transition techniques by using 6to4 and NAPT-PT mechanism to provide transparency between v4 and v6 end-to-end solution. Especially, they have implemented and verified functionality of their implementation of Session Initiation Protocol-Application Level Gateway (SIP-ALG) to work with their Transition Box. It is useful for allowing applications using SIP to pass through NAPT-PT.

Vieira, el. al., staff of Laboratories Over Next Generation Networks [21]-[22] have implemented and evaluated many types of transition mechanism including NAT-PT with various scenarios. They described and discussed each mechanism in each scenario which is useful for decision when any networks need to deploy IPv6 into their environments. They have presented the configurations of each technique based on MS Window, Linux, *BSD(FreeBSD, NetBSD, OpenBSD). Several service application e.g. HTTP, FTP, IRC(Internet Relay Chat), News, Mail, LDAP(Lightweight Directory Access Protocol) and DNS have been tested. They have also experimented with real users.

NAT-PT has been implemented, tested and deployed in the real environment at University of Southampton and University of Lancaster for testing. Both tests were called Bermuda2 [23], IPv6 transitioning project. They implemented other transition mechanisms such as tunnel brokers, 6to4 service.

Apart from these, network should be aware of security when NAT-PT is used. About the security on NAT-PT, Person, et al., presented this effect at the European ATC [24]-[25] test infrastructure. An IPv6 firewall providing the same features as IPv4 was used to filter and prevent unauthenticated connections from outside v6 network.

Cisco [26] deployed NAT-PT as a function of its device router. It tested and implemented as a gateway to operate the address mapping function. The similar works have been done by, Laboratories Over Next Generation Networks. It applied NAT-PT function and implemented on a computer router which based on Linux [27] operating system.

NAT-PT can be deployed to be a function of the border gateway which will become a single point of failure if the NAT-PT router goes down. This problem leads to have several NAT-

PT boxes on the same network. Multiple NAT-PT or mNAT-PT and load-balancing functions has been offered in the IETF internet draft [28]-[29]. It was designed for the large size of v6 networks. Several NAT-PT boxes are located at the border network and set to be a group. Cluster load-monitor as an extension function of mNAT-PT collaborates with DNS-ALG to manage load-balancing and assign IPv6 prefix to these NAT-PT boxes.

Lin and Sirisena presented a method to solve the limitations and undefined scenarios in SIIT and NAT-PT such as the use of private IPv4 addresses to communicate to IPv6, the communication between IPv4 and IPv6 nodes on the same network. They defined Extended Network Address & Protocol Translator (xNAPT) [30] as a proxy DNS server to allocate the appropriate address for translation between (private and global) IPv4 and IPv6. The translation is achieved through the use of DNS record. The study has shown that xNAPT allows IPv4 and IPv6 hosts to communicate with each other transparently.

In the lower of network layer, the processors have been applied to provide IPv4/IPv6 transition [31]. Grosse and Lakshman implemented middlebox on a network processor board and measure the performance of the hardware. They found that the network processors are capable of significant speedup of NAPT-PT and firewall processing compared to general-purpose processors.

3.4 Summary

This chapter explains the requirements to deploy IPv4/IPv6 transition mechanisms in the Internet. It presents an overview of the properties of each technique. The technique that was chosen to evaluate and enhance in this work is NAT-PT. Its outstanding feature is that it provides transparent communication and application between v4 and v6 networks without changing the end nodes. In addition, the principle of NAT-PT is similar to NAT which is already widely used.