CHAPTER 4

INVESTIGATION OF NAT-PT

This chapter presents the investigation of NAT-PT in detail. We start to explore NAT-PT by surveying the NAT-PT program. Then it is deployed and tested in a testbed network. NAT-PT basic functions and cooperation with ALG are the main part to be implemented. Several applications are used to test operations between v4 and v6 worlds in the laboratory for address and protocol translation. This chapter also presents an analysis of NAT-PT.

The survey of the NAT-PT program is presented in section 4.1. It is the existing implemented code for tasks of NAT-PT. The experimental design and setup network to determine NAT-PT properties are shown in section 4.2. Then the testing and result are in section 4.3. We conclude the testing and analyze NAT-PT features in section 4.4. The summary is in section 4.5.

4.1 Survey of NAT-PT program

The concept and principles of NAT-PT have already been presented in chapter 3. This part is practical to investigate and test NAT-PT. There are two main objectives: study NAT-PT code and implement into a testing network in order to discover a missing function for further capability enhancement.

NAT-PT was implemented as a patch file software for the kernel of the FreeBSD [32] operating system (OS) by the KAME project [33] which develops OS(s) and software to support IPv6. Basic tasks of NAT-PT were implemented as a function of the router implementation of FreeBSD to translate packet and addresses. These functions are protocol translation, address translation and cooperation with an ALG for address resolution. Each part is presented as follows:

4.1.1 Protocol translation

This function is used to translate IP and relevant protocols. Several modules were implemented to translate these protocols. They perform IP translation by changing the header of the packet. For the upper layer protocol, there are modules to adjust the header checksum of

TCP [15] and UDP [16] packets. They were implemented as functions to update pseudo header for TCP and UDP packets. A part of the protocol translation modules was created to perform port translation for TCP and UDP.

In addition, the relevant protocols such as ICMPv4 [14] and ICMPv6 [18] are necessary to be changed by the protocol translation so that v4 or v6 nodes can obtain error messages from the other domain, and to allow the exchange of informational messages, particularly ECHO and ECHO reply. NAT-PT does not perform protocol translation for fragmented ICMPv4 and ICMPv6 packets. The NAT-PT code drops a fragmented ICMP packet because the translator does not reassemble this fragmented packet.

4.1.2 Address translation

Address translation changes source and destination addresses to be the appropriate numbers in order to provide communication between IPv4 and IPv6 nodes. NAT-PT programs were implemented to perform address translation from IPv4 to IPv6 and vice versa relying upon the mapping rules. There are two directions of communications.

Firstly, for outbound session from an IPv6 to an IPv4 node, an IPv6 packet including its addresses must be translated to IPv4. In this case, NAT-PT requires a temporary IPv4 address to translate the IPv6 source address. A particular globally unique IPv4 address or a block of IPv4 addresses can be assigned in the address pool. The addresses used for this purpose must be such that packets sent to one of the addresses from any v4 host will be returned to the NAT-PT router.

The other direction, NAT-PT requires a temporary IPv6 address to translate IPv4. There is a simple way to generate a temporary IPv6 address. It is a combination of a unicast IPv6 prefix with 96 bits length and the unicast IPv4 address in the last 32 bits. Then the translated address looks like **IPv6-unicast-prefix::IPv4-address/96**. A network might have several NAT-PT routers as gateways in order to provide load balancing. These routers can be assigned the identical unicast IPv6 prefix.

A missing function of the NAT-PT program is that it was not implemented to translate multicast addresses. Then multicast packets are discarded. In fact, multicast IPv4 addresses do not have an address depletion problem. There are enough for current and expected future use. However, as the IPv4 unicast address space is almost exhausted, an eventual switch to IPv6 seems inevitable. Switching to IPv6 multicast with its even bigger group number space is a side effect of

the unicast switch.

Multicast applications have not been widely used in the Internet because they can consume significant network resources. However, some networks require and provide multicast services such as videoconference for meetings in a group of companies, or distance learning of a university between campus networks. The existing multicast applications are in v4 world and they use IPv4 to provide these services. If the network is to switch to IPv6 multicast must be included, including during the transition phase. Direct multicast communication between v4 and v6 worlds is not possible because they are different protocols. Thus we require a technique to provide multicast service between these worlds.

NAT-PT which performs addresses and protocol translation between IPv4 and IPv6 might be altered to do this work. However, the solution for this was not defined in the NAT-PT standard. No multicast address translation module is available in the NAT-PT program. It should be possible to enable multicast address translation between IPv4 and IPv6. The existing program must be extended for multicast.

In order to do that, we need to consider disciplines of multicast communication. And we need to know the way to add the new function into the existing system, the problem of extension and the required elements for NAT-PT with multicast.

4.1.3 Address discovery

DNS-ALG [13] was created to adjust IP addresses in DNS reply messages to be the appropriate format for the node making the query. When that node gets the modified destination address from the DNS-ALG, it can use the modified address to connect to the destination even though that destination does share its network layer protocol or address format with the original node. The ALG was implemented in a program named Trick or Treat Daemon (totd) [35] which was created to perform address mapping from IPv4 to IPv6 only. The original program allows v6 nodes to access servers in the v4 world given knowledge only of the server's name.

However, totd does not have interface to cooperate with the NAT-PT function. The ALG is unable to add the mapped address to the NAT-PT translation rules. The mapping list which is used by NAT-PT to decide to translate IP address is configured manually. For mapping v4 addresses to v6 this is not a problem, the transformation is entirely algorithmic – by configuring both NAT-PT and totd with the same value to use as the IPv6 prefix for this translation they can each be left to

operate independently while still achieving consistent results. The totd program was not implemented to provide address mapping from IPv6 to IPv4. When a v4 node wants to connect to a v6 node and resolves name to DNS server, there is no answer. Then a v4 node cannot create a connection to a v6 node by using its name. We assume this lack is primarily due to the requirement for communication between NAT-PT and the ALG in this case, and the current lack of any adequate API to allow this communication.

NAT-PT requires a new API to allow interoperation with the ALG. The principle of DNS-ALG to perform IPv6 to IPv4 address mapping which is presented in section 3.2.4.3 could be implemented in the original code to add the missing functionality, but this will not be useful until a mechanism to communicate the results to NAT-PT exists.

4.2 Test deployment of NAT-PT

In this section, we implement NAT-PT in network of our laboratory in order to determine operations and find out limitations of NAT-PT. The investigation is implemented according to the assumption as follows:

1. The testing is implemented in a simple testbed network
2. Only one NAT-PT box is deployed as a gateway in the network

We start to implement NAT-PT in a simple network which has only one NAT-PT router because the simple network makes easy to investigate tasks and look for limitations of NAT-PT. We do not need to implement NAT-PT in a complex network if we can deploy and discover its features from a basic environment. On the other hand, if the simple network cannot achieve to our objectives then we will adjust the environment to be an appropriate network.

4.2.1 Network design

The test network is configured in accordance with the assumptions of section 4.2 It is an isolated network that is setup inside our laboratory. The network consists of one IPv4-only and one IPv6-only network. A NAT-PT router acts as the gateway between the networks. The testbed network environment is shown in Figure 4.1. Each network has its own properties as follows:

IPv4-only network

Every node inside the v4 network is installed and configured with IPv4 addresses only. Only IPv4 applications run on this network. Any unicast IPv4 addresses could be chosen for this work – the address can deliver a packet to the correct recipient. They can be either private or global IPv4 addresses which makes no difference for IPv4 and IPv6 address translation.

IPv6-only network

Every node inside v6 networks is configured with IPv6 address only. They provide IPv6 applications only. Every node uses IPv6 global scope to communicate.

NAT-PT router

NAT-PT is configured as a function of the border gateway at the boundary between the v4 and v6 networks. All incoming and outgoing packets between the v4 and v6 realms must pass through the router. NAT-PT processes original and translated packets like a normal router. It looks for destination address in each packet and finds an appropriate path to forward it. We use a static routing table for unicast forwarding, the network is too simple to require dynamic routing.

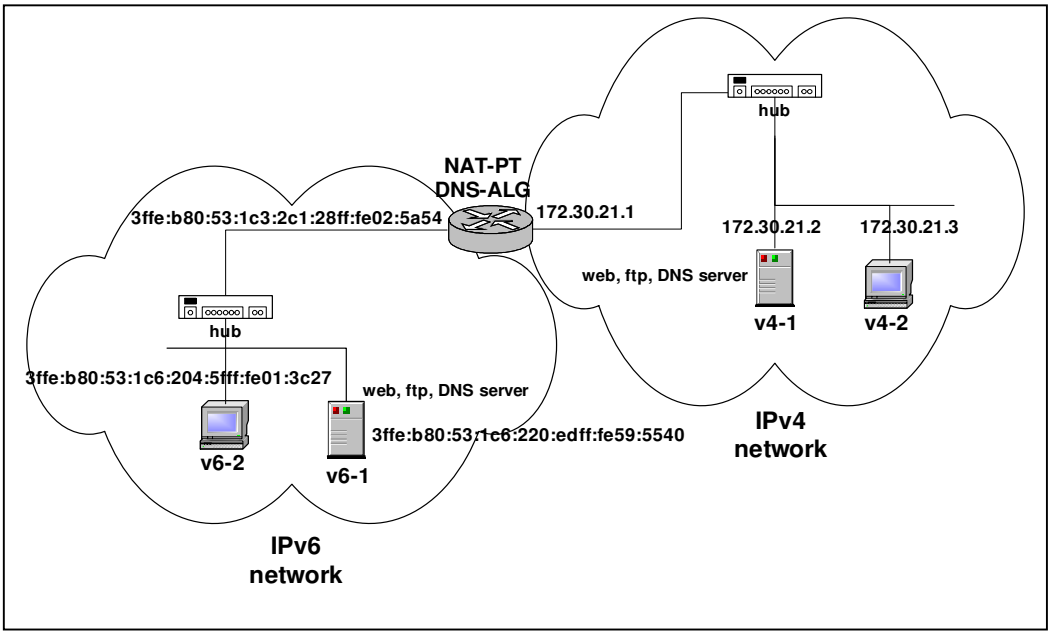

Figure 4.1 Testbed netwok environment

4.2.2 Network configuration

In this experiment, we configure IPv4 and IPv6 addresses for each network and assign addresses in the pool for translation as follows:

1. IPv4 network has been configured with network address 172.30.21.0/24.
2. For IPv6 network, IPv6 prefix number 3ffe:b80:53:1c6::/64 has been assigned to all v6 nodes in this site.
3. The pool for NAT-PT has been allocated with IPv4 block 10.10.25.0/24 and the IPv6 prefix for translation is 3ffe:501:41c:c1ad::/64

4.2.3 Equipment and software

Since the NAT-PT program was implemented on FreeBSD with kernel patch file from the KAME project, this investigation selected the same operating system. We choose FreeBSD-4.9-release to be the operating system of the NAT-PT router and every node in the v4 and v6 networks.

We want a generator to create packet for communication between v4 and v6 nodes in order to test protocol and address translation. We use several applications which use different application protocols such as Apache, SSH, ping, ping6, telnet, sendmail, and ftp in our testing. We also use a multicast application in our testing. It is the vic program, Videoconference tool.

4.3 Testing and result

The investigation has two main parts. Firstly, NAT-PT is tested to perform protocol and address translation from IPv6 to IPv4 in scenario 1. Translation in the other direction is tested in scenario 2. The DNS-ALG (totd) runs on the NAT-PT router both scenarios.All test use the network environment as shown in Figure 4.1. The objectives and procedures of testing are presented for each scenario as follows:

Scenario 1: Address translation from IPv6 to IPv4

This scenario is a communication from a v6 node in the v6 network to a v4 node in the v4 network. The objectives in this testing are as follows:

1. To test NAT-PT operation to perform protocol and address translation from IPv6 to

IPv4.

2. To test port translation.
3. To verify and validate the translation by using application programs and the mapping table.

Conditions of testing: This scenario is to test address translation from IPv6 to IPv4. Communication is initiated from any v6 node in the v6 network to a v4 node in the v4 network. Every v6 node connects to the same v4 node. We use applications and the mapping rules to verify NAT-PT operation.

When a v6 node wants to connect to a v4 node, it requires a destination address to connect to the v4 node. The destination address must be IPv6. We assign a unicast IPv6 prefix to be used to contain the entire IPv4 address space. The way to generate the temporary IPv6 address which maps the desired IPv4 address is presented in section 3.2.4.2.

In order to allow the connection from v6 to v4, the IPv6 packet and addresses must be changed to IPv4. The destination address translation has no options, the IPv4 address generated must identify the intended v4 destination node. The source address however can be anything, all that is required is that packets to the address chosen be returned to the NAT-PT router so they can be converted back to v6, and that the mapping chosen be stable for at least the life of a connection. The NAT-PT router is configured with a pool of IPv4 addresses to use for this purpose. We assign only one IPv4 address in the address pool. The single IPv4 address is shared and mapped to every IPv6 address of v6 nodes for outgoing session. The source port of every v6 node is mapped to the port which is registered in the rules.

The mapping rules are configured as in Figure 4.1. The IPv6 address of every v6 source node must be mapped to the single IPv4 address 10.10.25.48 whereas the source port must be changed to a value from the range 20000 to 30000. Different source ports identify different connections, perhaps from different source nodes.

```
prefix 3ffe:501:41c:c1ad::/64
map from any6 to 10.10.25.48 port 20000-30000
```

Figure 4.1 The mapping rules for IPv6 to IPv4 address translation

Note that this configuration style is not required. We could have allocated a block of IPv4 addresses, sufficient to allocate one to each v6 node, and not required port translation. We choose this configuration as it is a close match to real world NAT and probably NAT-PT, deployment, where the number of available global IPv4 addresses is very limited.

Procedure of testing: We create a connection from any v6 node to the v4 node with address 172.30.21.2. We use application program as in section 4.2.3 to test address and protocol translation. These applications are based on several protocols such as HTTP, ICMPv4/v6, SMTP, RTP/RTCP and FTP. We apply the same procedures to test every application. The following steps are the processes of NAT-PT to perform address translation from IPv6 to IPv4.

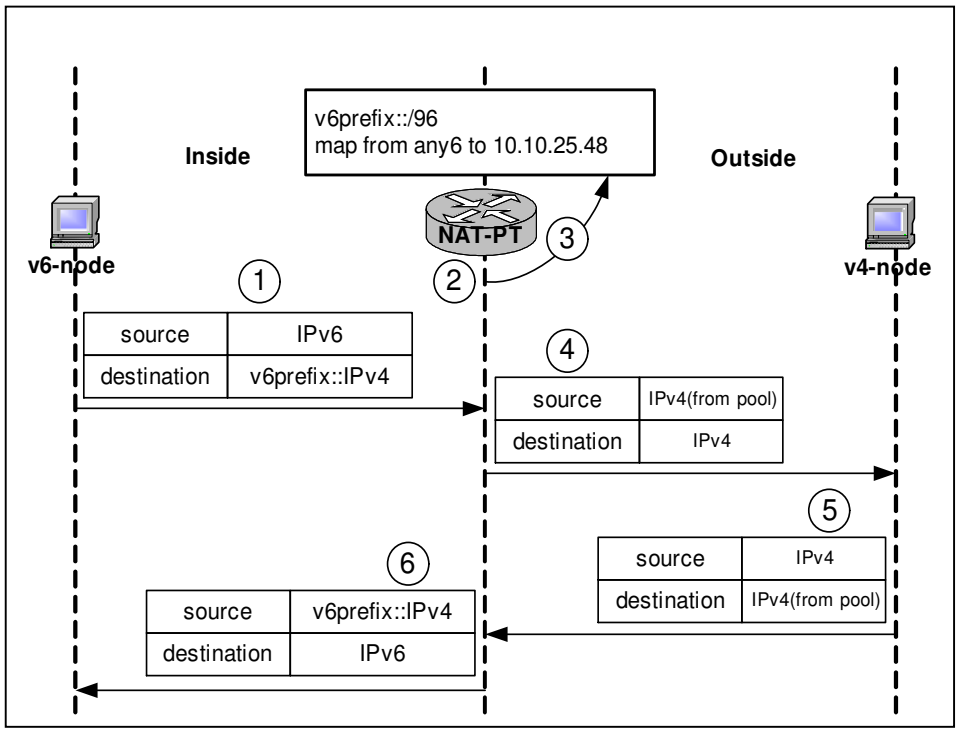

Figure 4.2 Procedures of NAT-PT to provide communication from v6 to v4 node

1. The v6 source node runs application and uses the destination address which is configured in the mapping rules as its destination number. Then it sends v6 packets to the v6 address selected. The packet to this address will be forwarded to the NAT-PT router.

2. The NAT-PT router, which is as the gateway of the v6 network, gets the v6 packets and looks for the destination address in the mapping rules to check whether it must perform address translation or not. If the destination address exists in the rules, it is translated.
3. Since the destination address has already mapped in the mapping rules, NAT-PT performs address translation from IPv6 to IPv4 following the rules. It translates the v6 packet to v4 by changing header of the packet, and mapping the v6 source address to a v4 address.
4. The translated packets are sent to the v4 network by the NAT-PT router.
5. When the v4 node replies to the v6 node, it sends v4 packets to the v4 address supplied by the NAT-PT router to map the original IPv6 source address.
6. NAT-PT gets the v4 packets and performs address translation from IPv4 to IPv6 in the reverse path.

Result of testing: NAT-PT successfully translates headers and addresses according to the mapping rules. The transport headers are correctly updated. ICMP packets are translated between IPvv4 and IPv6 formats. IPv6 nodes are able to communicate with the v4 node using any of the unicast applications mentioned section 4.2.3.

An example result is in Figure 4.3. A v6 wanted to access the web server from the v4 node. The v6 source addresses 3ffe:b80:53:1c6:203:6dff:fe16:ad50 connected to the v4 node by using the IPv6 destination address 3ffe:501:41c:c1ad::ac1e:1502. The IPv6 source and destination addresses were mapped to the IPv4 addresses 10.10.25.48 and 172.30.21.2 respectively. The source port is mapped from 1038 to 28686 but the destination port is not changed.

Local Address(src)	Local Address(dst)	Remote Address(src)	Remote Address(dst)
3ffe:b80:53:1c6:203:6dff:fe16:ad50.1038	3ffe:501:41c:c1ad::ac1e:1502.80	10.10.25.48.28686	172.30.21.2.80

Figure 4.3 Example of address mapping from IPv6 to IPv4

However, some applications cannot be provided by NAT-PT only for – example DNS and FTP. We want to know how NAT-PT operates when it performs translation of a packet that has IP

addresses in a part of its payload. For this we use the bind (DNS) and ftp programs to create this kind of packet.

When the v6 node wants to know the IP address of a node to which it wants to connect, including v4 nodes, it sends a DNS query packet to find the (v6) address to the DNS server for the domain being sought. If the server is a v4 server, which it is likely to be if the destination is on a v4 network, the query packet is sent via the NAT-PT router to the v4 DNS server but the v6 node gets the answer that there is no associated IPv6 address for the host name. For this scenario, the destination is an IPv4 node configured with IPv4 address only in the DNS server. The v6 node does not get an appropriate IP address of the destination and it cannot create a connection to that destination node. Although the DNS answer packets pass through the NAT-PT router, the exchange does not have the desired effect because there is no IP address in this payload.

Since DNS is an application layer protocol whereas NAT-PT is a protocol in transport layer, they are not visible and cannot communicate directly to each other. NAT-PT does not see addresses in the payload of a DNS packet and cannot perform address translation. To provide using application between v4 and v6 worlds, the query type in a DNS query, and the address in the DNS answer must be adjusted to be correct for the destination network. DNS and NAT-PT are required to have a gateway to provide communication to each other. For DNS, the DNS-ALG is used to map an appropriate destination address in the content of DNS packet for a source node requires that destination address.

For this scenario, IPv6 address of the v4 destination is not available in DNS server. The DNS-ALG arranges a temporary IPv6 address (which identify that v4 destination) as the destination address and sends that address to the v6 source node. The way that the DNS-ALG uses to generate the temporary IPv6 address is shown in section 4.2.2. The temporary IPv6 address and the original IPv4 address must be recorded in the mapping rules of NAT-PT in order to allow NAT-PT to perform address translation. We must manually add the pair of the temporary and original addresses into the mapping list because the DNS-ALG code has no interface to communicate with NAT-PT. This task is a programming technique and it will be improved in future work.

FTP control connection includes an IPv4 address in the PORT command, and in the server's response to the PASV command. As part of the development of IPv6, new versions of those commands, EPRT and EPSV, were developed which can carry either IPv4 or IPv6, and in the

future perhaps other, address types – though usually only the port number is required. Unfortunately old IPv4 servers are unlikely to support EPRT or EPSV whereas an IPv6 only client (or server) cannot use the PORT or PASV commands.

To allow FTP to work in this environment an ALG is required, that can translate address formats in the new (extended) commands, and even possible convert between the new and old command.

When the ftp program is run on a v6 node and uses the EPRT command to connect to the v4 FTP server (the v4 node in the v4 network) pass through the NAT-PT router, the router does not see the command which is in the payload of the FTP request packet. The NAT-PT router cannot modify IPv6 address of the v6 client to be an appropriate address which visible to the v4 server.

NAT-PT is required to have an ALG to be a gateway between the application and network layers to provide address mapping. The FTP-ALG is used to perform this task and allow transparent connections for ftp applications between the v4 and v6 nodes. The ALG already exists in the NAT-PT program. When the v6 client originates an FTP connection to the v4 server, the IPv6 address of the v6 client are changed from 3ffe:b80:53:1c6:203:6dff:fe16:ad50 to IPv4 address 10.10.25.48. The ALG [13] translates the network protocol from IPv6 (AF#2) to IPv4 (AF#1). It also changes port number following the rules as configured in Figure 4.1. NAT-PT allows an IPv6 FTP client to connect to a server on v4 node.

The vic program is a multimedia application and can use both unicast and multicast addresses to communicate. It uses the RTP/RTCP protocols [34] to transmit data and control sessions. We run vic on a v6 node and use the mapped IPv6 address in the mapping list to identify this application to the v4 node in the v4 network. The v4 node can connect and use vic application on the v6 node via the NAT-PT router.

We also run vic on the same v6 node using an IPv6 multicast address to identify the session. In order to make sure that the multicast program is advertised, we run vic on the other v6 nodes and join the same v6 group address. These v6 nodes can receive and display the v6 session from the v6 sender node.

We want to know whether this session is sent to the v4 network or not. We map the IPv6 multicast group address to IPv4 and configure it into the mapping rules. However, the IPv6 address is not translated and the v6 packets are dropped when they arrive to the router. NAT-PT does not

perform multicast address translation or process multicast packets.

Scenario 2: Address translation from IPv4 to IPv6

This scenario tests communication from a v4 node in the v4 network to any of the v6 nodes in the v6 network. The objectives of this testing are as follows:

1. To test NAT-PT operation to translate addresses from IPv4 to IPv6.
2. To verify and validate the translation by using applications and the mapping table.

Conditions of testing: This scenario is to test address translation from IPv4 to IPv6. A connection is initiated by a v4 node to any v6 node in the v6 network. Only one v4 node uses several v6 applications from different v6 nodes.

When the v4 node wants to connect to the v6 node, it requires an IPv4 destination address. In this testing, we assign a block of IPv4 addresses to use as temporary replacements for the v6 destination nodes. The assigned address is allocated following the method in section 3.2.4.2.

The v4 packet and address must be changed to be the appropriate IPv6 version in order to create the connection to those v6 nodes. NAT-PT requires an IPv6 address to translate IPv4 source address. We assign a particular IPv6 unicast prefix, 3ffe:501:41c:c1ad::/96, for address translation. The way to translate the address is presented in section 3.2.4.2. The block of IPv4 addresses used to map IPv6 destination addresses is as shown in Figure 4.4. The rules are used to translate IPv4 to IPv6 without port mapping.

```
map  from  10.10.10.14  to  3ffe:b80:53:1c6:203:6dff:fe16:ad50
map  from  10.10.10.16  to  3ffe:b80:53:1c6:220:edff:fe59:5540
map  from  10.10.10.17  to  3ffe:b80:53:1c6:204:5fff:fe01:3c27
```

Figure 4.4 The mapping rules for IPv4 to IPv6 address translation

Procedure of testing: We create connections from the v4 node with address 172.30.21.2 to several v6 nodes in the v6 network. The applications and protocols which are tested in scenario 1 are used in this scenario. The steps of NAT-PT to process address translation are similar to the procedure in the previous scenario. However, some parts are adjusted as follows:

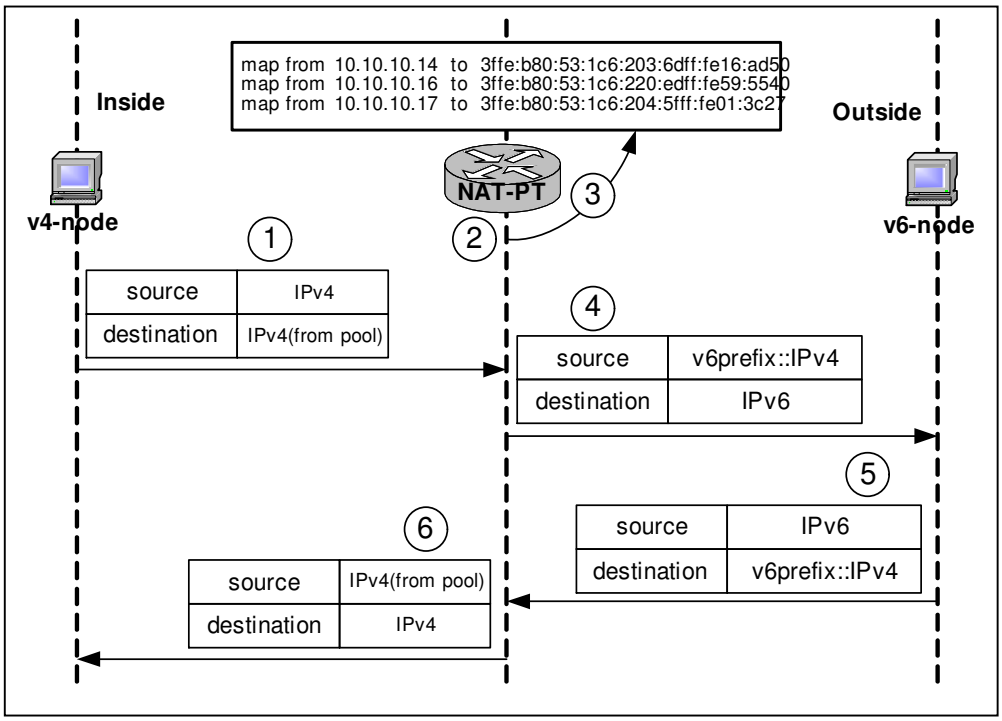

Figure 4.5 Procedures of NAT-PT to provide communication from v6 to v4 node

1. The v4 node runs an application and connects to a v6 node using a temporary IPv4 destination address which is in the mapping rules. It sends v4 packets to the temporary address which results in them being forwarded to the NAT-PT router.
2. The router gets the v4 packet and checks the destination address in order to perform address translation.
3. NAT-PT translates the packet by changing the header of the packet, and translates the IP addresses from IPv4 to IPv6.
4. Then the router sends the translated packet to the v6 network.
5. When the v6 node responds to the v4 node, it send v6 packets to the v6 address which appeared as the source address of the packet it received.
6. NAT-PT gets the v6 packets and translates the packet and IP addresses in the reverse direction following the mapping rules.

Result of testing: NAT-PT translates the protocol and addresses by changing packet header and mapping addresses from IPv4 to IPv6. The relevant fields in the header of v4 packet are updated and changed to be IPv6. Then the v4 node can access servers on the v6 network and communicate with any v6 node in the v6 network.

An example result is in Figure 4.6. The v4 node with address 172.30.21.2 runs the telnet program to connect to the destination node at address 10.10.10.14. When the connection arrives at the NAT-PT router, the source and destination addresses of the packet are translated to 3ffe:501:41c:c1ad::ac1e:1502 and 3ffe:b80:53:1c6:203:6dff:fe16:ad50 respectively. The source and destination ports (1163 and 23) are not mapped.

Local Address(src)	Local Address(dst)	Remote Address(src)	Remote Address(dst)
172.30.21.2.1163	10.10.10.14.23	3ffe:501:41c:c1ad::ac1e:1502.1163	3ffe:b80:53:1c6: 203:6dff:fe16:ad50.23

Figure 4.6 Example of address mapping from IPv4 to IPv6

When the v4 node wants to use an application, it must resolve the destination address by sending a DNS query to the DNS server. The query packet is used to find the domain and delivered to network where the DNS server contains that address. The packet is sent via the NAT-PT router to v6 node but the v4 node receives the answer that there is no associated IPv4 address for the host name. For this scenario, the destination is v6 node and it has IPv6 address only in the DNS server. The v4 node does not get a suitable IP address and cannot create a connection to that v6 node.

DNS application for IPv4 to IPv6 connection encounters to the same problem as the communication from IPv6 to IPv4. IP address in the DNS answer is not mapped to be an adequate address when the DNS packet passes through the NAT-PT router. NAT-PT cannot perform address translation in DNS payload. The DNS-ALG is used to provide address mapping to the address in the DNS payload. However, the DNS-ALG code (totd) does not implement the function to provide address mapping from IPv6 to IPv4. In testing, we manually map IP address of v6 destination node for DNS application and add that to the mapping list of NAT-PT.

We use an IPv4 address in NAT-PT's pool as a temporary address and maps to the IPv6

address of the v6 destination. We use the mapped IPv4 address as the destination address for the v4 source node as if the source node gets the destination address from the DNS server.

The ftp application for an FTP v4 client connects to an FTP v6 server has the same problem as the other case in scenario 1. For this, an FTP v6 server does not understand IPv4 address in an FTP request command, PORT, of an FTP v4 client. NAT-PT cannot perform address translation to that IPv4 address.

FTP-ALG is also required to provide address mapping in FTP commands which is in the part of FTP payload. When the v4 client originates ftp connection to the v6 server, the IPv4 address in the PORT command must be mapped to IPv6 and the command altered to EPRT. Source and destination IP addresses are mapped following the rules as defined in Figure 4.4.

For the vic program, the result is similar to the previous scenario, v6 to v4. NAT-PT does not translate multicast addresses and packets from IPv4 to IPv6 although we configure the mapped multicast addresses into the rules.

4.4 Conclusion and analysis of testing

In this section, we conclude and analyze features and operations of NAT-PT as the result of our experiments. The protocols that were used in our testing are summarized in Table 4.1.

We found that NAT-PT can provide protocol and address translation for unicast such as DNS, FTP, HTTP, SMTP, SSH, TFTP, ICMPv4, ICMPv6 and RTP/RTCP. However, like NAT, NAT-PT cannot perform its works to the protocols that contain IP addresses inside the payload of the packets such as DNS and FTP. It requires the ALGs to arrange the appropriate things in order to allow communication between NAT-PT and the application protocols. However, each application protocol has some differences in detail.

Table 4.1 Summary protocols in NAT-PT testing

Protocol	**NAT-PT**		**ALG**
	Inbound	**Outbound**	
DNS	Yes	Yes	Required
FTP	Yes	Yes	Required
HTTP	Yes	Yes	No
SMTP	Yes	Yes	No
SSH	Yes	Yes	No
TFTP	Yes	Yes	No
ICMPv4	ICMPv4	ICMPv6	No
ICMPv6	ICMPv6	ICMPv4	No
RTP/RTCP	Yes	Yes	No
Multicast	No	No	-

DNS is used to resolve a destination address for using applications in the Internet. NAT-PT can translate DNS packet from v4 to v6 and vice versa but it cannot translate IP address in the content of DNS payload. The ALG for DNS, DNS-ALG, is required to provide address mapping in the content of DNS payload, in order to visible and allow using applications in v4 to v6 world and vice versa.

FTP uses the IP address as a part of its commands to create a data connection from an FTP client to an FTP server. However, an FTP connection between two nodes that have different IP protocols cannot be established via the NAT-PT router. FTP commands are in the content of FTP payload and they are invisible from the NAT-PT router to perform address translation in the payload. For this, FTP-ALG is required to be a gateway to intercept and map IP address in the argument of FTP commands to be an appropriate address in order to create an FTP connection between these nodes.

Finally, we found that NAT-PT does not perform translation for multicast. Multicast packets are dropped and multicast group addresses are not translated. Thus multicast sessions

cannot be provided between v4 and v6 worlds via the NAT-PT router.

4.5 Summary

NAT-PT was implemented as a function of a router to perform protocol and address translation. However, it has some missing parts which require the extensions in programming and some new methods.

NAT-PT can provide application and communication between IPv4 and IPv6 networks for unicast protocols. It works well for static address configuration in the mapping rules. Some applications that contain IP address as a part of content or commands need the ALGs to provide the appropriate updates in order to allow connection between v4 and v6 nodes.

However, several limitations of NAT-PT have been explored. One of them is that NAT-PT does not provide multicast address translation and drops multicast packets. Then multicast sessions cannot be provided between the v4 and v6 worlds. In addition, the standard of NAT-PT did not define address and protocol translation for multicast.