Winter 12-10-2002

ENAT-PT: An Enhanced NAT-PT Model

Li-An Zeng

Zhao-Hui Cheng

Li Ling

Songrong Qian

Xu Liao

ENAT-PT: An Enhanced NAT-PT Model

Li-An Zeng, Zhao-Hui Cheng, Li Ling, Songrong Qian, Xu Liao

Institute of Information Science and Engineering
Fudan University, Shanghai, China
zenglian@sina.com

Abstract

NAT-PT would allow IPv4 nodes to communicate with IPv6 nodes transparently by translating the IPv6 address into a registered V4 address. However, NAT-PT would fall flat when the pool of V4 addresses is exhausted. NAPT-PT multiplexes the registered address' ports and will allow for a maximum of 63K outbound TCP and 63K UDP sessions per IPv4 address, but it is unidirectional. We present in this paper a novel solution ENAT-PT（an enhanced NAT-PT），which will allow for a great number of inbound sessions by using a single V4 address. By using ENAT-PT, we can visit V6 networks from a V4 network with a small address pool.

Key words: ENAT-PT; IPv6 Transition; NAT-PT; NAPT

1. Introduction

The key to a successful IPv6 transition is compatibility with the large installed base of IPv4 hosts and routers. Maintaining compatibility with IPv4 while deploying IPv6 will streamline the task of transitioning the Internet to IPv6. IPv6 transition mechanisms include providing complete implementations of both versions of the Internet Protocol[1,2], and tunneling IPv6 packets over IPv4 routing infrastructures[3,4]. They are designed to allow IPv6 nodes to maintain complete compatibility with IPv4, which should greatly simplify the deployment of IPv6 in the Internet, and facilitate the eventual transition of the entire Internet to IPv6. Another important and widely used technology is NAT-PT[5-9]. Network Address Translation is a method by which IP addresses are mapped from one address realm to another, providing transparent routing to end hosts. Protocol Translation define the translation rules between IPv4 headers and IPv6 headers[10]. Traditional NAT-PT such as NAPT-PT is unidirectional while Bi-Directional-NAT-PT allows IPv4-only node to visit IPv6-only node and vice versa, but Bi-Directional-NAT-PT can not reuse a registered address for more than one session. Thus, we develop ENAT-PT to solve this problem. The most outstanding feature of ENAT-PT is that ENAT-PT is bi-directional and it requires only a small pool of registered addresses.

The rest of this paper is organized as follows. Section 2 defines the frequently used terms in this paper. Section 3 introduces background knowledge such as NAT-PT and its variations; besides, we give the reason why ENAT-PT is proposed. Section 4 presents the principle of ENAT-PT model and describes its operation in detail. Section 5 analyzes the constraints of ENAT-PT model and discusses other considerations related to ENAT-PT. Section 6 concludes the paper.

2. Terms

Terms frequently used in this paper are defined as below. They may have special meaning in this paper.

Definition 1. ***V4 node, V6 node***: In this paper, we use the term V4 node to denote V4-only node, and V6 node to denote V6-only node[1].

Definition 2. ***Session***: A session is defined as the set of traffic that is managed as a unit for translation [11]. Sessions are uniquely identified by their ***session parameters.*** For TCP/UDP sessions, session parameters are the tuple of (*Sa, Sp, Da, Dp*); for ICMP query sessions, session parameters are the tuple of (*Sa, ICMP query ID, Da*). *Sa, Sp, Da, Dp* stand for source address, source port, destination address, destination port respectively.

Definition 3. ***Inbound session*** and ***outbound session***: A session flow indicates its direction in which the session is initiated with reference to a network interface [11]. In this paper, an inbound session flow is defined as a session flow initiated from a V4 node (to a V6 node), while an outbound session flow is initiated from a V6 node (to a V4 node).

Definition 4. ***NAT*** (Network Address Translation)**:** In this paper, NAT refers to translation of an IPv4 address into an IPv6 address and vice versa. While the V4 NAT [6,11] provides routing between private V4 and external V4 address realms, NAT in this paper provides routing between a V6 address realm and an external V4 address realm.

3. NAT-PT and Its Flavors

NAT-PT (Network Address Translation - Protocol Translation) is a standard track IETF RFC describing an IPv6/IPv4 translator. NAT-PT allows native IPv6 hosts and applications to communicate with native IPv4 hosts and applications, and vice versa. An NAT-PT device resides at the boundary between an IPv6 and IPv4 network. Each NAT-PT device retains a pool of globally routable IPv4 addresses which are used to assign to IPv6 nodes on a dynamic basis as sessions are initiated across the IPv6/IPv4 boundary. In addition to address translation, header translation is performed as described in the SIIT mechanism[10]. As opposed to SIIT which is a stateless translation mechanism, NAT-PT retains state via the IPv4 to IPv6 address mappings and which are retained for the duration of each session.

3.1 Traditional-NAT-PT (Outbound NAT-PT)

Traditional-NAT-PT would allow hosts within a V6 network to access hosts in the V4 network. In a traditional-NAT-PT, sessions are unidirectional, outbound from the V6 network. This is in contrast with Bi-directional NAT-PT, which permits sessions in both inbound and outbound directions. There are two variations to traditional-NAT-PT, namely **Basic-NAT-PT** and **NAPT-PT**.

3.1.1 Basic-NAT-PT

With Basic-NAT-PT, a block of V4 addresses are set aside for translating addresses of V6 hosts as they originate sessions to the V4 hosts in external domain. For packets outbound from the V6 domain, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For returned traffic, the destination IP address and the checksums as listed above are translated.

3.1.2 NAPT-PT

NAPT-PT extends the notion of translation one step further by also translating transport identifier (e.g., TCP and UDP port numbers, ICMP query identifiers). This allows the transport identifiers of a number of V6 hosts to be multiplexed into the transport identifiers of a single assigned V4 address. NAPT-PT allows a set of V6 hosts to share a single V4 address. NAPT-PT can be combined with Basic-NAT-PT so that a pool of external addresses are used in conjunction with port translation. NAPT-PT allows for a maximum of 63K outbound TCP and 63K UDP sessions per V4 address. For packets outbound from the V6 network, NAPT-PT would translate the source IP address, source transport identifier and related fields such as IP, TCP, UDP and ICMP header checksums. Transport identifier can be one of TCP/UDP port or ICMP query ID. For returned traffic, the destination IP address, destination transport identifier and the IP and transport header checksums are translated.

3.1.3 ALG

The NAT-PT translation device may additionally contain ALG's (Application Level Gateways). ALG's are necessary where IP addresses are embedded within the payload of an IP packet. For normal packet translation, NAT-PT would not look within the payload for IP addresses. For some applications where IP addresses may be embedded within the payload, an ALG is necessary to look inside the payload and translate those IP addresses.

3.2 Bi-Directional-NAT-PT

With Bi-directional-NAT-PT, sessions can be initiated from hosts in V4 network as well as the V6 network. V6 network addresses are bound to V4 addresses, statically or dynamically as connections are established in either direction. The name space (i.e., their Fully Qualified Domain Names) between hosts in V4 and V6 networks is assumed to be end-to-end unique. Hosts in V4 realm access V6-realm hosts by using DNS for address resolution.

A DNS-ALG[12,13] must be employed in conjunction with Bi-Directional-NAT-PT to facilitate name to address mapping. Specifically, the DNS-ALG must be capable of translating V6 addresses in DNS queries and responses into their V4-address bindings, and vice versa, as DNS packets traverse between V6 and V4 realms.

3.3 Challenge

NAPT-PT allows a set of V6 hosts to share a single V4 address, but NAPT-PT is unidirectional and applicable only for outbound sessions. With Bi-directional-NAT-PT model, however, every registered address is bound to a single V6 address. Once the address pool is exhausted, V4 nodes cannot establish sessions with other V6 nodes anymore. Thus there brings a challenge: how to establish a great number of inbound sessions by using a single V4 address ? ENAT-PT, which stands for Enhanced NAT-PT, will provide a better solution.

4. ENAT-PT

4.1 ENAT-PT Overview

ENAT-PT comprises of three parts: ENAT, NAPT and PT, see table 1.

The following of this paper mainly focuses on ENAT. For more information about NAPT and PT, please refer to references [5-11].

Table 1 ENAT-PT model components

Session Direction	Address Translation Method	Header Translation Method
Outbound	NAPT	PT
Inbound	ENAT	PT

With Bi-directional-NAT-PT, every registered address is bound to a single V6 address when a DNS query is performed and released when the session is terminated. With ENAT, a registered address bound to a V6 address would be released at the open of a session instead of the session's termination. Thus a registered address may be reused for another session just after the establishment of a session. Destination address (or source address of returned traffic) of subsequent packets of a session would be translated to the correct translated address according to their session parameters, which are retained in a session table.

4.2 ENAT Operation

Like NAT-PT, an ENAT-PT device resides at the boundary between an IPv6 and IPv4 network. Each ENAT-PT device retains a pool of globally routable IPv4 addresses (or registered addresses) which are used to assign to IPv6 nodes on a dynamic basis as sessions are initiated across the IPv6/IPv4 boundary. Besides, the ENAT-PT should maintain two tables, namely session table and binding table. The session table is as follows.

V4 Address	V4 Port	V6 Address	V6 Port	Translation Address

V4(V6) port here denotes the port associated to the V4(V6) address. There is an entry in the session table for each session.

The binding table is as follows.

Registered Address	Binding Address	State

There is an entry for each registered address in the binding table. The state of a registered address may be *free*, *bound*, *occupied*, or *occupied and bound*. See definition 5 below.

Definition 5. The state of a registered address:
Free: the initial state.
Bound: when the ENAT-PT bind it to a V6 host.
Occupied: when a registered address is used by some session, i.e., it's in the session table.
Occupied and bound: Occupied by one or more sessions, and, at the same time, bound to a V6 address.

The ENAT operation is as follows:

1.A DNS query should be performed before a session is established. This is done by the facilitation of the DNS-ALG, which is naturally embedded in an ENAT-PT. The DNS-ALG translates the DNS query and the ENAT-PT forwards it to the DNS server. The ENAT-PT will choose a registered address whose state is not *bound* (or *occupied and bound*) and bind it to the V6 address returned by the DNS server. The address binding will be added to the binding table. ENA-PT returns the registered address as the translation address of the destination V6 node. This is very similar to the binding procedure in NAT-PT. If the state of the registered address is *free*, it should be changed to *bound;* if it's *occupied*, it should be changed to *Occupied and bound* after the binding.

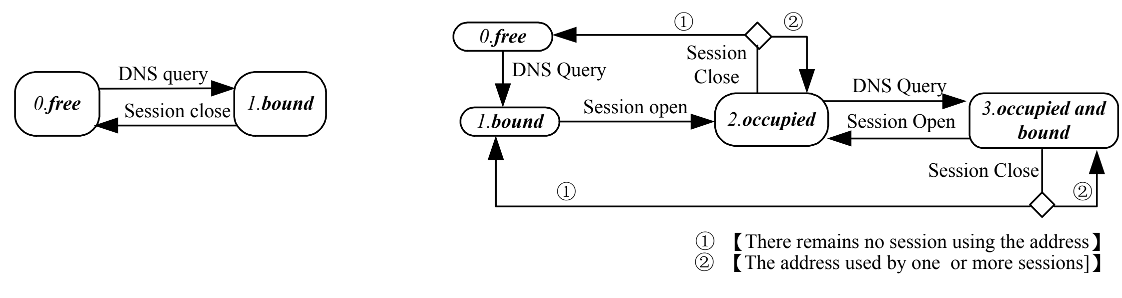
2. The ENAT-PT captures and intercepts the first packet of the session. Destination address of the packet is translated to the V6 address by querying the binding table. An entry, which contains the IP of V4 node, port associated to the V4 address, IP of V6 node, port associated to the V6 address and the translation address, will be added to the session table. Then the state of the registered address is changed to *occupied* and the address binding will be cleared in the binding table.

3. The ENAT-PT captures and intercepts subsequent packets of the session and translates their addresses by querying the session table.

Fig.1-(b) is the state diagram of a registered address with ENAT. It shows how does ENAT differ from NAT(see Fig 1-(a) as a contrast). Note that when a session is opened, a registered address is released and can be bound to another V6 address.

Fig.2 is an example to illustrate the operation of ENAT. ***Q1*** and ***Q2*** are V4 nodes while ***H1*** and ***H2*** are V6 nodes. There are two registered addresses in the binding table. The registered address ***R0*** is statically bound to the DNS server. The initial state of ***R1*** is *free*, see table 2. The ENAT-PT will intercept, capture and translate the packets. The session table is empty.

For convenience, we use ***Q1***.address to denote the IP address of ***Q1***, and ***R1***.state to denote the state of Registered address ***R1***…etc.



(a) The state of a registered address with NAT

(b) The state of a registered address with ENAT

Fig.1 The state of a registered address

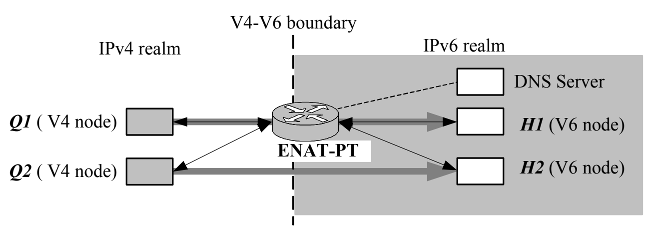


Fig.2 An illustration for ENAT

Table 2 Binding table (0)

Registered address	Bound Address	State
R0	DNS-Server.address	*Bound*
R1	-	*Free*

Assume that V4 node ***Q1*** initiates a session to V6 node ***H1***. The DNS server returns ***R1*** as the translation address of ***H1***, thus ***R1*** is bound to ***H1***.address now, the binding table is as follows (see table 3).

Table 3 Binding table (1)

Registered address	Bound Address	State
R0	DNS-Server.address	*Bound*
R1	***H1***.address	*Bound*

When the session is open, the first packet of the session is delivered from ***Q1*** to ***H1*** whose session parameters are:

Sa = ***Q1****.address*,***Sp*** = *p1*, ***Da*** = ***R1***,***Dp*** = *p2* (1)

Then we add the session parameters to the session table (see table 4) and change ***R1***.state to *occupied*, and the binding table is as table 5.

Table 4 Session table (1)

V4 Address	V4 Port	V6 Address	V6 Port	Translation Address
Q1.address	*p1*	***H1***.address	*p2*	***R1***

Table 5 Binding table (2)

Registered address	Bound Address	State
R0	DNS-Server.address	*Bound*
R1	-	*Occupied*

Address of packets of the session is translated according to the session table. For packets from ***Q1*** to ***H1***, the translation as follows:

*(****Q1****.address,p1,****R1****,p2)* → *(PREFIX::****Q1****.address, p1, ****H1****.address, p2)* (2)

for returned traffic, the translation is:

*(****H1****.address, p2, PREFIX:****Q1****.address, p1)* → *(* ***R1****, p2, ****Q1****.address, p1)* (3)

NOTE: The prefix PREFIX::/96 is advertised in the stub domain by the ENAT-PT device, and packets addressed to this *PREFIX* in the stub domain will be routed to the ENAT-PT device. The pre-configured PREFIX only needs to be routable within the IPv6 stub domain and as such it can be any routable prefix that the network administrator chooses [8].

Table 6 Binding table (3)

Registered address	Bound Address	State
R0	DNS-Server.address	*Bound*
R1	***H2***.address	*Occupied and Bound*

Once a registered address is released (when the session the registered address associated to is opened), it can be bound to another V6 address. For example, another V4 node ***Q2*** initiates a DNS query (e.g., resolving the name of node ***H2***) just after the establishment of the previous session. ENAT-PT will bind ***R1*** to ***H2***.address as translation address of ***H2***, see Table 6.

So session parameters of packets from ***Q2*** to ***H2*** may be:

Sa = ***Q2****.address*, ***Sp*** = *p3*, ***Da*** = ***R1***, ***Dp*** = *p4*. (4)

After the session is opened, the session table is as table 7 and the binding table is as table 5.

Table 7 Session table (2)

V4 Address	V4 Port	V6 Address	V6 Port	Translation Address
Q1.address	*p1*	***H1***.address	*p2*	***R1***
Q2.address	*p3*	***H2***.address	*p4*	***R1***

For packets from ***Q2*** to ***H2***, the translation is as follows:

*(****Q2****.address,p3,****R1****,p4)*→*(PREFIX::****Q2****.address,p3, ****H2****.address, p4)* (5)

for returned traffic, the translation is:

*(****H2****.address, p4, PREFIX:****Q2****.address, p3)* → *(* ***R1****, p4, ****Q2****.address, p3)* (6)

The above deals with TCP/UDP sessions. For ICMP sessions, the operation is pretty much the same thing, but sessions are identified by their source/destination address and ICMP query ID.

4.3 ENAT Analysis

(1) When the state of a registered address is *bound,* it can not be bound to another V6 address. The number of sessions that can be set up at the same time by using a single registered address is depended on two factors: T_D, which is the average duration of a registered address from *bound* (or *occupied and bound*) state to *occupied* state, and T_S, the average duration of a session.

$$N_{max} = (T_S + T_D) / T_D \quad (7)$$

(2) In table 7, if ***Q1***.address = ***Q2***.address (i.e., ***Q1*** = ***Q2***) and *p1* = *p3* and *p2* = *p4*, session parameters of the two sessions will be identical (see formula (1) and (4)). The ENAT-PT needs more information to tell one session from the other. In this case, the ENAT-PT should reject the connect request of the second session to avoid confusion. For TCP sessions, the ENAT-PT may discard the SYN & ~ACK packet, however, there is no deterministic way of recognizing the start of a non-TCP session. Fortunately, most operating systems will select different source ports (for the client) for different sessions.

(3) ENAT and NAPT are both essential parts of ENAT-PT model. ENAT deals with inbound sessions and NAPT deals with outbound sessions. If we use the different registered address pools for inbound and outbound sessions, ENAT and NAPT need no modification. However, ENAT and NAPT can share the same registered addresses pool and the same session table, but the session table should be changed to the following form because NAPT will change the clients' ports.

V4 Address	V4 Port	V6 Address	V6 Port	Translation Address	Translation port

5. Related Considerations

5.1 The start and Termination of Sessions [11]

The first packet of every TCP session may be recognized by the presence of SYN bit and absence of ACK bit in the TCP flags. There is no deterministic way of recognizing the start of a non-TCP session. A heuristic approach would be to assume the first packet with hitherto non-existent session parameters as indicating the start of a new session. The end of a TCP session is detected when FIN is acknowledged by both halves of the session or when either half receives a segment with the RST bit in TCP flags field. However, because packets may be dropped or retransmitted, TCP sessions can be assumed to have been terminated only after a period of time subsequent to this detection. In addition, it is necessary for an ENAT-PT to clean up unused state about TCP sessions that no longer exist. In the case of non-TCP sessions, session timeouts must be configurable because they vary greatly from application to application. Another way to handle session terminations is to timestamp entries and retire the longest idle session when it becomes necessary .

5.2 Limitations[8,11]

Here are the most important limitations with ENAT-PT. They are associated with NAT-PT as well.

(1) Applications depending on global address may work improperly because a registered address may be bound to different V6 addresses at the different time or even at the same time, and vice versa.

(2) An ENAT-PT will not translate IP-address within packet payload thus applications such as SNMP, FTP will not work properly. An SNMP-ALG [14] or an FTP-ALG [15] is necessary for theses applications.

(3) A DNS query must be performed before every session opens[8].

5.3 Security

Transport layer security techniques such as TLS [16], TCP MD5 Signature Option [17] can work properly because ENAT does not change port numbers. Security mechanisms of IP layer such as AH [18] protect the packet from address modifications and will not work.

5.4 Fragmentation

Although ENAT-PT does not translate transport identifiers, it has to know the port in a packet before translating the address and transmitting the packet, thus it is necessary to track record of IP fragments[19].

6. Conclusion

We have implemented a prototype of ENAT-PT on Linux based on BT Labs' NAT-PT implementation [20] and succeeded to initiate multi sessions from V4 nodes to different V6 nodes with a single V4 registered address. ENAT combined with NAPT-PT will provide a simple and effective solution for intercommunications between V4 nodes and V6 nodes with a small address pool. This will greatly reduce the consumption of IPv4 registered addresses for IPv4 networks to communicate with IPv6 network.

Reference

[1] R. Gilligan, E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC2893, August 2000.

[2] K. Tsuchiya, H. Higuchi, "Y. Atarashi, Dual Stack Hosts using the Bump-In-the-Stack Technique (BIS)", RFC2767,February 2000.

[3] P. Fasano, I. Guardini, D. Lento, "IPv6 Tunnel Broker A. Durand", RFC3053, January 2001.

[4] D. Haskin, "Routing Aspects of IPv6 Transition", Callon, RFC2185, September 1997.

[5] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines", RFC3235, January 2002.

[6] P. Srisuresh, K. Egevang, "Traditional IP Network Address Translator (Traditional NAT) ", RFC3022, January 2001.

[7] T. Hain, "Architectural Implications of NAT", RFC2993, November 2000.

[8] G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT) ", RFC2766, February 2000.

[9] R. Bush, "Delegation of IP6.ARPA (Updates RFC2874, RFC2772, RFC2766, RFC2553, RFC1886) ", RFC3152, August 2001

[10] E. Nordmark, "Stateless IP/ICMP Translation Algorithm (SIIT) ", RFC2765, February 2000.

[11] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[12] G. Tsirtsis, P. Akkiraju, A. Heffernan, "DNS extensions to Network Address Translators (DNS_ALG)", FRC2694, September 1999.

[13] Thomson, S., and C. Huitema. "DNS Extensions to support IP version 6", RFC 1886, December 1995.

[14] D. Raz, J. Schoenwaelder, B. Sugla, "An SNMP Application Level Gateway for Payload Address Translation", RFC2692, October 2000.

[15] Allman, M., Ostermann, S. and C. Metz, "FTP Extensions for IPv6 and NATs", RFC 2428, September 1998.

[16] T. Dierks,C. Allen, "The TLS Protocol Version 1.0", RFC2246 January 1999.

[17] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[18] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[19] Postel, J. (ed.), "Internet Protocol-DARPA Internet Program Protocol Specification", RFC 791, September 1981.

[20] Linux-based Userspace NAT-PT available at http://www.ipv6.or.kr/english/download.htm