Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: March 29, 2009

H. Miyata
M. Endo
Yokogawa Electric Corp.

September 29, 2008

sNAT-PT: Simplified Network Address Translation - Protocol Translation draft-miyata-v6ops-snatpt-02

Status of this Memo

This Internet-Draft will expire on March 29, 2009.

Copyright Notice

Abstract

This document specifies an IPv4-to-IPv6 transition mechanism to provide accessibility for IPv6 node to IPv4 node, and vice-versa. The goal of this document is not providing the most fundamental technology which could works well with additional technology.
We used to have an technology called NAT-PT[RFC2766]. NAT-PT was designed to work with problematic DNS Application Level Gateway. So, it was changed to historical state by [RFC4966].
This document attempts to simplify NAT-PT specification, removing dependability on Application Layer Gateway as well as resolving problems pointed in [RFC4966].

Table of Contents

1. Introduction

Disclaimer:
This proposal is incomplete. It is posted to seek comments on plausibility and to represent the approach how to simplify the NAT-PT which was mixed up with ALGs.

IPv6 is a new version of the Internet Protocol(IP) designed to improve IPv4 to allow for future Internet growth.
Recently, it is predicted that the IPv4 address would be run out in several years and IPv6 would be deployed.
On the other hand, IPv4 has been maintained to extend its lifetime.
So, the transition period is predicted to be long. During the period, IPv6 node need to co-exist with IPv4 node.
During the period, there would be some node which can use only IPv4 (IPv4 Only Node) as well as node which can use only IPv6(IPv6 Only Node). And they will need to communicate.
So, the translation technology is required.

There are some standarized translation technologies,
"Stateless IP/ICMP Translation Algorithm"(SIIT)[RFC2765] and
"An IPv6-to-IPv4 Transport Relay Translator"(TRT)[RFC3142].

SIIT has a big advantage that does not require to maintain the session state in Translator Box.
But SIIT is designed to be used by small IPv6 network. And it requires IPv6 host to implement functionalities to be assinged IPv4 address.

TRT has simple architecture and it does not require any modification for both IPv6 and IPv4 host.
But TRT Translatoion Box is designed to translate in transport layer.
To provide communication between IPv6 Only Node and IPv4 Only Node, it establishs two sessions, one is with IPv6 Only Node, another is with IPv4 Only Node. So it is costful. Also it is designed to provide only TCP sessions initiated by IPv6 Only Node.

NAT-PT was designed to provide both TCP and UDP communications.
Also it attempted to provide communications initiated by both IPv6 Only Node and IPv4 Only Node.
NAT-PT does not require IPv6 node and IPv4 node to implement special functionalities to communicate via Translation Box.
In 2007 it was depricated by some reasons described in [RFC4966].
But its basic and simple translation behavior is helpful for some situations, which does not expect perfect translation.

This documents attmept to recycle NAT-PT specification removing dependability on specific application and resolving some issues listed in [RFC4966].

2. Terminology

The majority of terms used in this document are borrowed almost as is from [RFC2663]. The following lists terms specific to this document.

2.1 Network Address Translation (NAT)

The term NAT in this document is very similar to the IPv4 NAT described in [RFC2663], but is not identical. IPv4 NAT translates one IPv4 address into another IPv4 address. In this document, NAT refers to translation of an IPv4 address into an IPv6 address and vice versa.

While the IPv4 NAT [RFC2663] provides routing between private IPv4 and external IPv4 address realms, NAT in this document provides routing between a IPv6 address realm and an external IPv4 address realm.

2.2 NAT-PT flavors

Just as there are various flavors identified with IPv4 NAT in [RFC2663], the following NAT-PT variations may be identified in this document.

2.2.1 Traditional NAT-PT

Traditional-NAT-PT would allow hosts within a IPv6 network to access hosts in the IPv4 network. In a traditional-NAT-PT, sessions are uni-directional, outbound from the IPv6 network. This is in contrast with Bi-directional-NAT-PT, which permits sessions in both inbound and outbound directions.
Just as with IPv4 traditional-NAT, there are two variations to traditional-NAT-PT, namely Basic-NAT-PT and NAPT-PT.

With Basic-NAT-PT, a block of IPv4 addresses are set aside for translating addresses of IPv6 hosts as they originate sessions to the IPv4 hosts in external domain. For packets outbound from the IPv6 domain, the source IP address and related fields such as IP, TCP, UDP and ICMP header checksums are translated. For inbound packets, the destination IP address and the checksums as listed above are translated.

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 IPv6 hosts to be multiplexed into the transport identifiers of a single mapped IPv4 address. NAPT-PT allows a set of IPv6 hosts to share a single IPv4 address. Note that NAPT-PT can be combined with Basic-NAT-PT so that a pool of external addresses are used in conjunction with port translation.

For packets outbound from the IPv6 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 inbound packets, the destination IP address, destination transport identifier and the IP and transport header checksums are translated.

2.2.2 Bi-Directional-NAT-PT

With Bi-directional-NAT-PT, sessions can be initiated from hosts in IPv4 network as well as the IPv6 network. IPv6 network addresses are bound to IPv4 addresses statically or dynamically.
For dynamic address binding Application Level Gateway(ALG)[RFC2663] is required.
There should be various kinds of ALG. The detail specification of ALG is out of scope of this document.

Bi-directional-NAT-PT which maps one IPv4 address to one IPv6 address should be called Basic-Bi-directional-NAT-PT.
Port-Mapping maps a pair of IPv4 address and TCP or UDP port to a pair of IPv6 address and TCP or UDP port.

2.3 Protocol Translation (PT)

PT in this document refers to the translation of an IPv4 packet into a semantically equivalent IPv6 packet and vice versa. Protocol translation details are described in [RFC2765].

2.4 Application Level Gateway (ALG)

ALG is an application specific agent that allows a IPv6 node to communicate with a IPv4 node and vice versa.
Some applications carry network addresses in payloads. But NAT-PT is application unaware and does not snoop the payload. ALG would snoop the payload to modify and configure the NAT-PT gateway dynamically if required.
ALG could work in conjunction with NAT-PT to provide support for many kind of such applications.

2.5 Dummy Prefix

The IPv6 Prefix to map IPv4 address to IPv6 address. And the length is 96 bit.
In this document Dummy Prefix is represented as PREFIX or PREFIX::/96 when IPv6 address or prefix is described.
The prefix PREFIX::/96 is advertised in the IPv6 network by the NAT-PT gateway, and packets addressed to this PREFIX will be routed to the NAT-PT gateway. The pre-configured PREFIX only needs to be routable within the IPv6 network and as such it can be any routable prefix that the network administrator chooses.

2.6 Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Conceptual Model

This document attempts to separate ALGs form NAT-PT gateway logically as described in Fig. 3.1.
NAT-PT gateway and ALG can colocate physically.
Some ALG can be implemented in Translation Engine if required.

This chapter makes use of internal conceptual variables to describe the behavior of NAT-PT gateway.
The specific variable names are just an example. An implementation is not required to have them in the exact form described here, so long as its external behavior is consistent with that described in this document.

One of the most important point here is that "Address Mapping Table" and "Translation Rule Table" is the most fundamental point that define the behavior how to translate the packet regardless how the entries are generated.

Another point is that the NAT-PT gateway MUST maintain the address mapping, translation rule or session status to recycle the IPv4 address and TCP/UDP ports. Also, to send ICMP Error Messages to appropriate destination, the information of translated session is important to map the destination address.

```
                      +------------------------+
                      |           ALG          |
                      | e.g., DNS-ALG, SIP-ALG |
                      +------------------------+
                           |   ^
                           |   |
                 +---------|---|------------------------+
             +-> |         v   |                        |
     Config. |   |   +-----------+ +-----+              |
       I/F  -+   |   |   ALG-IF  | | CUI |              |
      (*4)   |   |   +-----------+ +-----+              |
             +-> |       |      |  /  |                 |
                 |       |      | /   |                 |
                 |       |      |/    |                 |
                 |       |      |     |                 |
                 |       |     /|     |                 |
                 |       |    / |     |                 |
             +-> |       |   /  |     |                 |
             |   |       v  v   v     v                 |
             |   | +---------+ +---------+  +---------+ |
             |   | |AAtbl(*1)| |TRtbl(*2)|  |TStbl(*3)| |
             |   | +---------+ +---------+  +---------+ |
    Target   |   |        ^      ^           ^          |
      of    -+   |        |      |           |          |
     This    |   |        |      |           |          |
   Document  |   |      +----------------------+        |
             |   |      |  Translation Engine  |        |
             |   |      +----------------------+        |
             |   |                                      |
             |   |           NAT-PT gateway             |
             +-> +--------------------------------------+
```

Fig. 3.1 Conceptual Model
(*1) Address Mapping Table
(*2) Translation Rule Table
(*3) Session Status Table
(*4) Configuration Interface

3.1 Address Mapping Table

To translate the packet, An IPv4 address must be mapped to an IPv6 address regardless the direction of the session.
So, the address mapping must be defined in this table.

This table must have following informations.

IPv6 Address
The IPv6 address to which IPv4 address be mapped.

IPv4 Address
The IPv4 address which is mapped to above IPv6 address.
If the Translation Type of correspondent entry is NAPT-PT, the specified address can be shared with other entry.

Translation Type
The type of translation must be defined.
e.g., NAPT-PT or NAT-PT

Stability
The stability type of this mapping. This is used for recycle purpose. This value specifies this mapping is "static" or "dynamic". If it is mapped dynamically, and there are no session entry on "Session Status Table" associated to this entry for a while, the entry and associated entry in "Translation Rule Table" MUST be removed for recycle purpose.

3.2 Translation Rule Table

The NAT-PT gateway needs to know how to handle the packet. So, this table defines the rule what kind of packet must be handled in what kind of manner.

This table must have following informations.

Address Mapping Table Entry
The entry number of Address Mapping Table which is used by this entry.

Direction
The allowed direction of session initiation is defined.
e.g.,

```
IPv6-to-IPv4 (IPv6 node initiate the session)
IPv4-to-IPv6 (IPv4 node initiate the session)
Bi-dir       (Bi-direction)
```

IPv6 Address

The IPv6 address of end-node which can use this entry.
If the Direction value of correspondent entry is IPv6-to-IPv4, the address can be specified by range as well as wildcard.
e.g.,

```
2001:DB8:b:a::/64
2001:DB8:b:a:1234:5678:9abc:def0/128
any
```

IPv4 Address

The IPv4 address of end-node which can use this entry.
If the Direction value of correspondent entry is IPv4-to-IPv6, the address can be specified by range as well as wildcard.
e.g.,

```
192.0.2.0/24
192.0.2.4/32
any
```

Protocol

The Protocol which is allowed to translate.
e.g.,
TCP, UDP, ICMP

3.3 Session Status Table

The entries of this table MUST be generated dynamically when the session is initiated. And it MUST be maintained and removed according to the session status.
The correspondent session entry is generated after examining the "Translation Rule Table", if the session is allowed.

In TCP session:
The entry is generated by SYN packet. The entry is deleted when session is closed.
In UDP session:
The entry is generated by first packet. The entry is deleted after pre-configured time from last packet.

Protocol

The protocol of this entry.
e.g., TCP, UDP, ICMP

IPv6 Node

The IPv6 address of IPv6 side end node of correspondent session.
It can not be specified by range or wildcard.

IPv4 Node

The IPv4 address of IPv4 side end node of correspondent session.
It can not be specified by range or wildcard.

Port on IPv6 Node
The Port number of TCP or UDP, which is used on IPv6 side end node for this session.

Port on IPv4 Node
The Port number of TCP or UDP, which is used on IPv4 side end node for this session.

Mapped IPv6 Address
The address which is mapped to the address specified in "IPv4 Node" field. It must be the synthesis address.
The packet addressed to this address MUST be routed to the NAT-PT gateway.

Mapped IPv4 Address
The address which is mapped to the address specified in "IPv6 Node" field.
The packet addressed to this address MUST be routed to the NAT-PT gateway.

Port on Gateway IPv6 Side
The Port number of TCP or UDP, which is used on NAT-PT gateway for this session on IPv6 side.

Port on Gateway IPv4 Side
The Port number of TCP or UDP, which is used on NAT-PT gateway for this session on IPv4 side.

Remaining Time
Typically in UDP session, the entry MUST be removed when it is not in use. This field stores the remaining time for its expiration. After each packet translation, NAT-PT gateway must reset to the pre-configured value. The value MUST be decreased by aging method.

3.4 Configuration Interface

"Address Mapping Table" and "Translation Rule Table" MUST be configured
statically or dynamically.
So, the configuration Interface is required.
The administrator would configure static rule. So, the User Interface (A.K.A. GUI, CUI) would be used.
Some kinds of ALG, (e.g.. DNS-ALG, FTP-ALG) would generate dynamic rule, so the interface to set the rules are required.

The Interface is TBD

4. Dummy Prefix Definition

Each NAT-PT gateway MUST be pre-configured a Dummy Prefix. The Dummy Prefix assigned to individual NAT-PT gateway MUST be unique. The uniqueness MUST be guaranteed within the IPv6 network they attached. Backup model and Load balance model is out of scope of this document. The packet addressed to the synthesis IPv6 address MUST arrive to the associated NAT-PT gateway.

The format is described in Fig. 4.1.

```
                                                           1
             1         2       6         7   9             2
   0123456789012345678901234...01234567890...01234567890...012345678
   +------------------------//-----+------//-------+----//---------+
   |            IPv6 Prefix        |     IDENT     | IPv4 Address  |
   |               64 bit          |     32 bit    |     32 bit    |
   +------------------------//-----+------//-------+----//---------+
   |                                               |
   |<-----------------Dummy Prefix---------------->|
```

Fig. 4.1 Dummy Prefix Format

The length of Dummy Prefix is 96-bit. The top 64-bit is a routable unicast prefix of IPv6. Any prefix can be assigned by the administrator from the address space he/her is administrating as far as it is not assigned.

The following 32-bit, represented as IDENT MUST be an value assigned by IANA to indicate that translator would exist in the path.
More requirement for IDENT is described in Chapter 13.

The Dummy Prefix MUST be followed by encoded IPv4 address.

NOTE: As [ID-baker] mentioned if the Dummy Prefix less matches to the address assigned to the IPv6 client, it chooses the non-synthetic address naturally, Assigning the Dummy Prefix range from the range which is far from actual used unicast range.
Moreover, considering the usage inside the enterprise (See the third NOTE in Chapter 12, Applicability Statement), if the Dummy Prefix is selected from the prefix assigned to the enterprise, the client would prefer synthetic address than native address. To avoide this situation, using a prefix which less matchs to the enterprise prefix is useful.
But assigning a prefix to each enterprise will significantly increase the routing table. This is difficult trade-off.
When using DNS rewriting service, the client will not receive both synthetic and native address as far as DNS service attempt

to resolve the native service first.
Or, if the IPv6 network is stub network, well-known address which is far from commonly used unicast address area would be helpful.

5. NAT-PT Operation

NAT-PT offers a straight forward solution based on transparent routing [RFC2663] and address/protocol translation, allowing a large number of applications in IPv6 and IPv4 realms to inter-operate without requiring any changes to these applications.

5.1 Traditional NAT-PT (IPv6 to IPv4)

In the following paragraphs we describe the operation of traditional-NAT-PT and the way that connections can be initiated from a host in IPv6 domain to a host in IPv4 domain through a traditional-NAT-PT

5.1.1 Basic NAT-PT Operation

```
      (IPv6-B)-+
               |                  +===============+
      (IPv6-A)-+-(NAT-PT)---------| IPv4 network |--(IPv4-C)
                    |             +===============+
             (pool of IPv4 addresses)
```

Figure 4.1: IPv6 to IPv4 communication

```
       Node IPv6-A has an IPv6 address -> 2001:DB8:b:a::7654:3210
       Node IPv6-B has an IPv6 address -> 2001:DB8:b:a::7654:3211
       Node IPv4-C has an IPv4 address -> 192.0.2.12
```

NAT-PT has a pool of addresses including the IPv4 subnet 10.0.0.0/24

The IPv4 addresses in the address pool could be allocated one-to-one to the IPv6 addresses of the IPv6 end nodes in which case one needs as many IPv4 addresses as IPv6 end points. In this document we assume that the IPv6 network has less IPv4 addresses than IPv6 end nodes and thus dynamic address allocation is required for at least some of them.
Say the IPv6 Node IPv6-A wants to communicate with the IPv4 Node IPv4-C.
IPv6-A creates a packet with:

SA=2001:DB8:b:a::7654:3210 and
DA=PREFIX::192.0.2.12

The packet is routed via the NAT-PT gateway, where it is translated to IPv4.

If the outgoing packet is not a session initialisation packet, the NAT-PT SHOULD already have stored some state about the related session, including mapped IPv4 address and other parameters for the translation. If this state does not exist, the packet SHOULD be silently discarded.

If the packet is a session initialisation packet, the NAT-PT locally allocates an address (e.g: 10.0.0.10) from its pool of addresses and the packet is translated to IPv4. The translation parameters are cached for the duration of the session and the IPv6 to IPv4 mapping is retained by NAT-PT.

The resulting IPv4 packet has SA=10.0.0.10 and DA=192.0.2.12. Any returning traffic will be recognised as belonging to the same session by NAT-PT. NAT-PT will use the state information to translate the packet, and the resulting addresses will be SA=PREFIX::192.0.2.12, DA=2001:DB8:b:a::7654:3210. Note that this packet can now be routed inside the IPv6-only stub network as normal.

5.1.2 NAPT-PT Operation

NAPT-PT, which stands for "Network Address Port Translation + Protocol Translation", would allow IPv6 nodes to communicate with the IPv4 nodes transparently using a single IPv4 address. The TCP/UDP ports of the IPv6 nodes are translated into TCP/UDP ports of the registered IPv4 address.

While NAT-PT support is limited to TCP, UDP and other port multiplexing type of applications, NAPT-PT solves a problem that is inherent with NAT-PT. That is, NAT-PT would fall flat when the pool of IPv4 addresses mapped for translation purposes is exhausted. Once the address pool is exhausted, newer IPv6 nodes cannot establish sessions with the outside world anymore. NAPT-PT, on the other hand, will allow for a maximum of 63K TCP and 63K UDP sessions per IPv4 address before having no TCP and UDP ports left to map.

To modify the example sited in figure 4.1, we could have NAPT-PT on the border router (instead of NAT-PT) and all IPv6 addresses could be mapped to a single IPv4 address 10.0.0.10.

IPv6 Node IPv6-A would establish a TCP session with the IPv4 Node IPv4-C as follows:

IPv6-A creates a packet with:

```
SA=2001:DB8:b:a::7654:3210 , source TCP port = 3017 and
DA=PREFIX::192.0.2.12, destination TCP port = 23.
```

When the packet reaches the NAPT-PT box, NAPT-PT would map one of the TCP ports from the mapped IPv4 address to translate the tuple of (Source Address, Source TCP port) as follows:

```
SA=10.0.0.10, source TCP port = 1025  and
DA=192.0.2.12, destination TCP port = 23.
```

The returning traffic from 192.0.2.12, TCP port 23 will be recognized as belonging to the same session and will be translated back to IPv6 as follows:

```
SA=PREFIX::192.0.2.12, source TCP port = 23;
DA=2001:DB8:b:a::7654:3210 , destination TCP port = 3017
```

5.2 Bi-Directional-NAT-PT

5.2.1 Basic Bi-Directional-NAT-PT Operation

To provide incoming session, IPv4 address MUST be mapped one-to-one to IPv6 address. The mapping could be done via configuration interface.
In this section the description is based on the situation that mapped address and translation rule are already configured.

NAT-PT has a pool of addresses including the IPv4 subnet 10.0.0.0/24.

Say the IPv4-C wants to communicate with the IPv6-A.
Somehow 10.0.0.10 is mapped to the IPv6-A, and IPv4-C knows the mapped address. (ALG is expected to map the address and inform the address to IPv4-C dynamically.)
Then, the IPv4-C creates a packet with:

```
SA=192.0.2.12        and
DA=10.0.0.10
```

The packet is routed via the NAT-PT gateway, where it is translated to IPv6.

If the incoming packet is not a session initialisation packet, the NAT-PT SHOULD already have stored some state about the related session, including mapped IPv4 address and other parameters for the translation. If this state does not exist, the packet SHOULD be silently discarded.

If the packet is a session initialisation packet, the NAT-PT locally search the IPv6 address associated to 10.0.0.10 (2001:DB8:b:a::7654:3210) from its "Address Mapping Table" and the packet is translated to IPv6. The translation parameters are cached for the duration of the session and the IPv6 to IPv6 mapping is retained by NAT-PT.

The resulting IPv6 packet has SA=PREFIX::192.0.2.12 and DA=2001:DB8:b:a::7654:3210.
Any returning traffic will be recognised as belonging to the same session by NAT-PT. NAT-PT will use the state information to translate the packet, and the resulting addresses will be as follows.
SA=10.0.0.10, DA=192.0.2.12.

5.2.2 Port-Mapping Operation

Basic Bi-directional-NAT-PT maps one IPv4 address to one IPv6 address.
Port-Mapping requires to map one pair of IPv4 address and port to one pair of IPv6 address and port.

Say the IPv4-C wants to access to http service on the IPv6-A. Somehow IPv4 address and TCP port pair (10.0.0.10, 30080) are mapped to the IPv6 address and TCP port pair(2001:DB8:b:a::7654:3210, 80), and IPv4-C knows the mapped pair. Actually, the method how to inform the mapped information is out of scope of this document. But several method could be considered. for example, the administrator of the server advertise it to the users manually(most primitive method). or if the application uses some protocol to negotiate the session initiation, the proxy of the application can configure the mapping to translator and inform it to client application.
Then, the IPv4-C creates a packet with:

```
SA=192.0.2.12, source TCP port = 1025
DA=10.0.0.10, destination TCP port = 30080
```

The packet is routed via the NAT-PT gateway, where it is translated to IPv6. The translated packet is;

```
SA=PREFIX::192.0.2.12, source TCP port = 1025
DA=2001:DB8:b:a::7654:3210, destination TCP port = 80
```

The returning traffic from 2001:DB8:b:a::7654:3210, TCP port 80 will be recognized as belonging to the same session and will be translated back to IPv6 as follows:

```
SA=2001:DB8:b:a::7654:3210 , source TCP port = 80
DA=PREFIX::192.0.2.12, destination TCP port = 1025
```

And it should be translated as follows:

```
SA=10.0.0.10, source TCP port = 30080
```

```
      DA=192.0.2.12, destination TCP port = 1025
```

6. IPv6 Address mapping

An IPv6 address MUST be mapped to an IPv4 address by prepending Dummy Prefix to the IPv4 address.
The format MUST be as [PREFIX]::[IPv4 address].
The address can be automatically calculated.

7. IPv4 Address mapping

IPv4 address mapping can be done by either statically or dynamically.
To map dynamically some ALGs are required.
The ALG could be DNS-ALG, SIP-ALG, FTP-ALG etc...
The specification of ALG is various, it depends on application.
This document is attempting to remove the dependency on specific application. So the specifications of ALGs are out of scope of this document.

The borderline between ALG and NAT-PT gateway is "Address Mapping Table" and "Translation Rule Table".

This documents describes the behavior after the entry in both tables are set.
For the NAT-PT gateway, the behavior for both static entry and dynamic one are almost same.

Only the difference is expiration of the entry based on the value of Stability field of "Address Mapping Table".
If the value of Stability field indicate Dynamic, the entries in these tables MUST be maintained as described in chapter 3.

8. Protocol Translation Details

The IPv4 and ICMPv4 headers are similar to their IPv6 counterparts but a number of field are either missing, have different meaning or

different length. NAT-PT SHOULD translate all IP/ICMP headers from IPv4 to IPv6 and vice versa in order to make end-to-end IPv6 to IPv4 communication possible. Due to the address translation function and possible port multiplexing, NAT-PT SHOULD also make appropriate adjustments to the upper layer protocol (TCP/UDP) headers. Some application requires ALG to complete its communication. FTP-ALG, for example, would make to FTP payload as an FTP packet traverses from IPv4 to IPv6 realm or vice versa. But any kind of ALG is out of scope of this document.

Protocol Translation details are described in [RFC2765], but there are some modifications required to SIIT because of the fact that NAT-PT also performs Network Address Translation.

8.1 Translating IPv4 headers to IPv6 headers

This is done exactly the same as in SIIT apart from the following fields:

Source Address:
The low-order 32 bits is the IPv4 source address. The high-order 96 bits is the designated PREFIX for all IPv4 communications. Addresses using this PREFIX will be routed to the NAT-PT gateway (PREFIX::/96)

Destination Address:
NAT-PT retains a mapping between the IPv4 destination address and the IPv6 address of the destination node. The IPv4 destination address is replaced by the IPv6 address retained in that mapping.

8.2 Translating IPv6 headers to IPv4 headers

This is done exactly the same as in SIIT apart from the Source Address which should be determined as follows:

Source Address:
The NAT-PT retains a mapping between the IPv6 source address and a mapped IPv4 address. The IPv6 source address is replaced by the IPv4 address retained in that mapping.

Destination Address:
The original IPv6 packets that are translated have a destination address of the form PREFIX::IPv4/96. Thus the low-order 32 bits of the IPv6 destination address is copied to the IPv4 destination address.

8.3 TCP/UDP/ICMP Checksum Update

NAT-PT retains mapping between IPv6 address and an IPv4 address. This mapping is used in the translation of packets that go through NAT-PT gateway.

The following sub-sections describe TCP/UDP/ICMP checksum update procedure in NAT-PT, as packets are translated from IPv4 to IPv6 and vice versa.

8.3.1 TCP/UDP/ICMP Checksum Update from IPv4 to IPv6

UDP checksums, when set to a non-zero value, and TCP checksum SHOULD be recalculated to reflect the address change from IPv4 to IPv6. The incremental checksum adjustment algorithm may be borrowed from [RFC3022].
In the case of NAPT-PT, TCP/UDP checksum should be adjusted to account for the address and TCP/UDP port changes, going from IPv4 to IPv6 address.

When the checksum of a IPv4 UDP packet is set to zero, NAT-PT MUST evaluate the checksum in its entirety for the IPv6-translated UDP packet. If a V4 UDP packet with a checksum of zero arrives in fragments, NAT-PT MUST await all the fragments until they can be assembled into a single non-fragmented packet and evaluate the checksum prior to forwarding the translated V6 UDP packet.

ICMPv6, unlike ICMPv4, uses a pseudo-header, just like UDP and TCP during checksum computation. As a result, when the ICMPv6 header checksum is computed [RFC2765], the checksum needs to be adjusted to account for the additional pseudo-header.

Note, there may also be adjustments required to the checksum due to changes in the source and destination addresses (and changes in TCP/UDP/ICMP identifiers in the case of NAPT-PT) of the payload carried within ICMP.

8.3.2 TCP/UDP/ICMP Checksum Update from IPv6 to IPv4

TCP and UDP checksums SHOULD be recalculated to reflect the address change from IPv6 to IPv4. The incremental checksum adjustment algorithm may be borrowed from [RFC3022]. In the case of NAPT-PT, TCP/UDP checksums should be adjusted to account for the address and TCP/UDP port changes, going from IPv6 to IPv4 addresses. For UDP packets, optionally, the checksum may simply be changed to zero.
The checksum calculation for a IPv4 ICMP header needs to be derived from the IPv6 ICMP header by running the checksum adjustment algorithm [RFC3022] to remove the IPv6 pseudo header from the computation.
Note, the adjustment must additionally take into account changes to the checksum as a result of updates to the source and destination addresses (and transport ports in the case of NAPT-PT) made to the payload carried within ICMP.

8.4 ICMP translation

The ICMP translation is described in [RFC2765], It is based on previous RFC for ICMPv6[RFC2463]. And it is obsoleted by [RFC4443]. So the description of ICMP translation needs to be revised.

8.4.1 ICMP translation from IPv4 to IPv6

There are no additional type and code for ICMPv4 after 2765.
So, no change is required.

8.4.2 ICMP translation from IPv6 to IPv4

[RFC4443] additionaly assigned the type 100, 101, 200 and 201. There is not common meaming of them, So, the translator SHOULD silently drop them. And the translator MAY have the interface to confiigure correspoondent IPv4 type and code only for private experimentation purpose.
Other added code should be dealed as follows.

```
Destination Unreachable (Type 1),
   Code 2 - Beyond scope of source address
            Set Code to 1 (host unreachable).
            This would happen by mis-configuration of Dummy Prefix.
            So, the translator SHOULD inform the issue to its
            administrator somehow.

   Code 5 - Source address failed ingress/egress policy
            Set Code to 1 (host unreachable).

   Code 6 - Reject route to destination
            Set Code to 1 (host unreachable).
```

9. Host Implementation

This specification does not require any additional function to use NAT-PT gateway. So current existing IPv6 devices can use NAT-PT gateway. However, if IPv6 devices implement some functionality it can resolve some issues listed on [RFC4966].

```
e.g., Client application can display dialog to indicate users that
      the there is a translator in the path.
      Server application detect the existence of translator in the
      path. Then it can select the behavior according to the
      pre-defined policy.
```

The detail behavior should be described in higher version or individual document.

NOTE: The method how to identify the synthetic address is shown in Chapter 4 and 13. [ID-bagnulo] also mentioned the another approach, DNS option namely SAS. Basically, from the layer point of view, those ideas are not so different. The better method should be discussed.

10. Application Layer Gateway support

Some applications contain IP address in payload of the packet to initiate a new session. In such case, the translation rule should be configured dynamically.
To allow this, ALG is required and the communication method between ALG and NAT-PT gateway is required.
NAT-PT gateway needs the information in both "Address Mapping Table" and "Translation Rule Table".

The protocol between NAT-PT gateway and ALG is TBD.

11. NAT-PT Limitations and Future Work

All limitations associated to NAT [RFC2663] are also associated to NAT-PT. Here are the most important of them in detail, as well as some unique to NAT-PT.

11.1 Load Balance

Once we separate ALGs from NAT-PT gateway, load balance is easily achieved. For smarter dynamic load balance, ALG and NAT-PT gateway SHOULD have communication method to share the load of NAT-PT gateways as described in [ID-endo].
Anyway the ALG will inform some informations to NAT-PT gateway to be stored in "Address Mapping Table" and "Translation Rule Table".
The purpose of this document is describing the behavior after those informations have been set. So load balance is out of the scope of this document.

11.2 Redundancy

The router redundancy technology is defined in [RFC3768]. It is called "Virtual Router Redundancy Protocol"(VRRP).
But it is not enough to provide redundancy, because NAT-PT gateway MUST maintain the state of each session and rule.
So, status synchronization technology would be required in addition to VRRP.
It is clear that the status synchronization technology MUST synchronize the information in all of "Address Mapping Table",

"Translation Rule Table" and "Session Status Table".
So, the redundancy can be provide by extending sNAT-PT.
But it is out of the scope of this document.

11.3 SCTP

SCTP is the useful protocol for the stable communications. If the two pathes use the different translator, the associations can be stable as expected. And it is possible by using different Dummy Prefix for each destination address.
On the other hand, SCTP is possible to add independetly from other tranport protocols. And as [ID-jennings] mentioned it is not the time to support it, since even IPv4-IPv4 NAT does not support it.
So, it should be considered later, considering the consistency of IPv4-IPv4 NAT.

11.4 Topology limitations

There are limitations to using the NAT-PT translation method. It is mandatory that all requests and responses pertaining to a session be routed via the same NAT-PT router. One way to guarantee this would be to have NAT-PT based on a border router that has unique Dummy Prefix to a stub domain, where all IP packets are either originated from the domain or destined to the domain.

11.5 Protocol Translation Limitations

A number of IPv4 fields have changed meaning in IPv6 and translation is not straightforward. For example, the option headers semantics and syntax have changed significantly in IPv6. Details of IPv4 to IPv6 Protocol Translation can be found in [RFC2765].

11.6 Impact of Address Translation

Since NAT-PT performs address translation, applications that carry the IP address in the higher layers will not work. In this case Application Layer Gateways (ALG) need to be incorporated to provide support for those applications. This is a generic problem with NAT and it is fully described in [RFC2663].

11.7 Lack of end-to-end security

One of the most important limitations of the NAT-PT proposal is the fact that end-to-end network layer security is not possible. Also transport and application layer security may not be possible for applications that carry IP addresses to the application layer. This is an inherent limitation of the Network Address Translation function.

Independent of NAT-PT, end-to-end IPSec security is not possible across different address realms. The two end-nodes that seek IPSec network level security must both support one of IPv4 or IPv6.

11.8 Multicast Translation

The packet translation of multicast is almost same as unicast case.
The Bi-directional-NAT-PT gives the hint of the method.
The multicast translation requires two mapping of destination multicast addresses and source addresses.

11.8.1 The translation of multicast from IPv6 to IPv4

The IPv4 multicast address MUST be assigned to the IPv6 multicast address somehow. Also, the IPv4 address MUST be assinged to the IPv6 address of the sendor. These mapping can be done manually at least. The well-known address like the addresses assigned to ntp serveice MUST be mapped to the correspondent addresses.

Following example shows the case when IPv6-A sends a multicast packet addressed to ff08::101. The address 224.0.1.1 MUST be assigned to ff08::101. and 10.0.0.10 must be assinged to 2001:DB8:b:a::7654:3210.

```
SA=2001:DB8:b:a::7654:3210        and
DA=ff08::101
```

The translator translates the packet as follows;

```
SA=10.0.0.10        and
DA=224.0.1.1
```

To make the packet addressed to the IPv6 multicast address reach the the translator, the translator MUST join for the correspondent address.
When the configuration was removed the translator must send leave as well.

11.8.2 The translation of multicast from IPv4 to IPv6

The IPv6 multicast address MUST be assigned to the IPv4 multicast address somehow. The IPv6 address to be assinged to the IPv4 address of the sendor is calucurated by prependig the Dummy Prefix to the original IPv4 address.
The well-known address like the addresses assigned to ntp server MUST be mapped to the correspondent addresses.
To make the packet addressed to the IPv4 multicast address reach the the translator, the translator MUST join for the correspondent address.

When the configuration was removed the translator must send leave as well.

11.9 Twice NAT-PT

Theoretically, it should work well.
Further investigation should be done later.

12. Applicability Statement

NAT-PT is a very useful tool to provide communication between IPv6 Only Node and IPv4 Only Node.

When only the communication which IPv6 Only Node initiate is required, NAPT-PT is useful because it requires few IPv4 address.

In this direction, the Translation engine can work separately from ALGs. But, as [ID-bagnulo] and [ID-endo] introduced, DNS-ALG like service, which provide synthetic address, should be helpful.
Such kind of service does not need to intercommunicate with Translation Engine. They just need to share Dummy Prefix.
i.e., The introduced Translation Engine can work with totd, which deos not have any interface to intercommunicate with Translation Engine. It is same as TRT. TRT itself can work alone, but using totd should be very helpful.

NOTE: This document takes the same approach as TRT, Basically the Translation Engine can work alone. Some kind of ALGs are considered as the utility services.

NOTE: Totd is a famous DNS proxy implementation, which can generate synthetic address using pre-configured Dummy Prefix. The packets, addressed to the synthetic address, arrive to the appropriate Translation Engine, which is pre-configured the same Dummy Prefix. Usually, totd attempts to provide non-synthetic address first. If it is impossible it attempts to provide synthetic address. Though it is designed to work with TRT, it works with sNATPT for the communication initiated by IPv6 Only node.

The most primitive usage is manual configuration. If the application allows to specify the address directly, the user can specify the synthetic address. Also, if an administrator wants to provide some services to IPv6 Only Node which works on IPv4 Only Node. The administrator can configure the actual DNS RR using synthetic address. This usage could occur when the Translation Engine is placed in front of the servers.

NOTE: There are two kinds of typical usage at least. One is placing the Translation Engine in front of the clients, around the entrance/exit of enterprise network. The other one is placing Translation Engine in front of the servers.

When the communication needs to be able to initiated by either IPv4 Only Node or IPv6 Only Node, Bi-Directional-NAT-PT is helpful.
The Bi-directional-NAT-PT can work with statically configured address mapping.

Also, when using Bi-Directional-NAT-PT, ALG should be helpful.
Some ALG can be implemented inside the Translation Engine(Fig. 3.1).
Some ALG can be implemented outside the Translation Engine. Moreover it can be implemented outside the NAT-PT gateway.

sNAT-PT provides very basic functions. So, it has some limitations when it works alone. But it can be helpful when it is combined with some other technologies.

13. IANA Considerations

The Dummy Prefix format is described in Fig. 4.1.
The 32-bit, represented as IDENT in Fig. 4.1, MUST be the value which indicates that this address is synthesis address.
IANA has assigned FE under their OUI(00-00-5E) to indicate that an IPv4 address is encoded in following 32-bit as represented in Fig.13.1.

According to the definition, it could be used for NAT-PT technology as the address contains the encoded IPv4 address.

```
0                       23      31                                63
|        OUI            |  0xFE |           IPv4 address          |
000000ug00000000 0101111011111110 xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx
```

Fig. 13.1 IPv4 embedded address Format

This address is used by ISATAP [RFC5214], one of tunneling technologies.

In RFC4966 it is stated that NAT-PT gateway existence in the path must be detected by the end-node, so same address SHOULD not used for NAT-PT. It is desired to assign another value to NAT-PT technology.

14. Security Considerations

Section 11.4 of this document states that end-to-end network and transport layer security are not possible when a session is intercepted by a NAT-PT. Also application layer security may not be possible for applications that carry IP addresses in the application layer.

Finally, all of the security considerations described in [RFC2663] are applicable to this document as well.

15. References

15.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3022] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC2765] Nordmark, E., "Stateless IP/ICMP Translator (SIIT)", RFC 2765, February 2000.

[RFC3768] R. Hinden, Ed, "Virtual Router Redundancy Protocol (VRRP)" , RFC 3768, April 2004.

[RFC3142] J. Hagino and K. Yamamoto, "An IPv6-to-IPv4 Transport Relay Translator", RFC 3142, June 2001.

[RFC5214] F. Templin, T. Gleeson and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)" , RFC 5214, March 2008.

[RFC4443] A. Conta, S. Deering, M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.

[RFC4966] C. Aoun, E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator", RFC 4966, July 2007.

15.2 Informative References

[RFC2463] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) specificagtion", RFC 2766, December 1998.

[RFC2766] G. Tsirtsis, P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.

[ID-bagnulo] M. Bangnulo, P. Matthews, I.van Beijnum,"NAT64/DNS64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-bagnulo-behave-nat64-00, June 2008.

[ID-endo] M. Endo, H. Miyata, "Translator Friendly DNS Proxy", draft-endo-v6ops-dnsproxy-00, August 2008.

[ID-baker] X. Li, C. Bao, F. Baker, "IVI Update to SIIT and NAT-PT", draft-baker-behave-ivi-00, September 2008.

[ID-jennings] C. Jennings, "NAT for IPv6-Only Hosts", draft-jennings-behave-nat6-00, July 2008.

Appendix A: Changes from draft-miyata-v6ops-snatpt-01 (Sep. 8, 2008)

- 2.2.2 Bi-Directional-NAT-PT was modified to support "Port-Mapping".

- Add "NOTE" in Chapter 4. to discuss the Dummy Prefix.

- In 5.1.2 NAT-PT Operation, "inbound NAPT-PT" description was removed.

- 5.2 Bi-Directional-NAT-PT was separated into "5.2.1 Basic-Bidirectional-NAT-PT" and "5.2.2 Port-Mapping".

- 5.2.2 Port-Mapping was added to improve the "inbound NAPT-PT"

- 11.3 SCTP, fill the description.

- 11.8 Multicast, fill the description.

- 12. Applicability Statement. the 3rd NOTE and the last sentence were improved.

- Correcting typos.

Appendix B: Changes from draft-miyata-v6ops-snatpt-00 (Feb. 1, 2008)

- In 5.2 Bi-directional NAT-PT, clarified the description on address mapping.Dynamic address mapping was removed.(Comment by Dan Wing)

- Improve the description of conceptual figure. Fig. 3.1.

- Add "NOTE" in Chapter 8. And based on it, add an informative reference.

- Enriched Chapter "12. Applicability Statement". Adding example.

- Correcting typos.

Author's Address

Hiroshi Miyata
Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi,
Tokyo, 180-8750
JAPAN

Email: h.miyata@jp.yokogawa.com

M. Endo
Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi,
Tokyo, 180-8750
JAPAN

Email: masahito.endou@jp.yokogawa.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Dan Wing gave me a comment to improve this document.