Host-based Translation Problem Statement

chengang@chinamobile.com

Why we need host based translation

- Two IP families need talk each other, otherwise there are totally separated two worlds;
- There exists IPv4 or IPv6 only network;
- How to support conventional IPv4 applications in IPv6 only network, IPhone store already has more than 60,000 applications.
- The implementation of operator’s service has been long-time running, quite stable, and hard to upgrade.
- Modify the host is very difficulty, but modify the host’s network stack is not that difficulty.

Possible way to go?

- Most hosts have dual stack already, and current translation solution consider only single stack:
  - We could use the other stack’s ability
  - Sending twice DNS queries (Separated A/AAAA records) isn’t an issue
  - Host is also possible to get both IPv6 prefix and IPv4 address assignment.
  - Operator customize the host more than before.

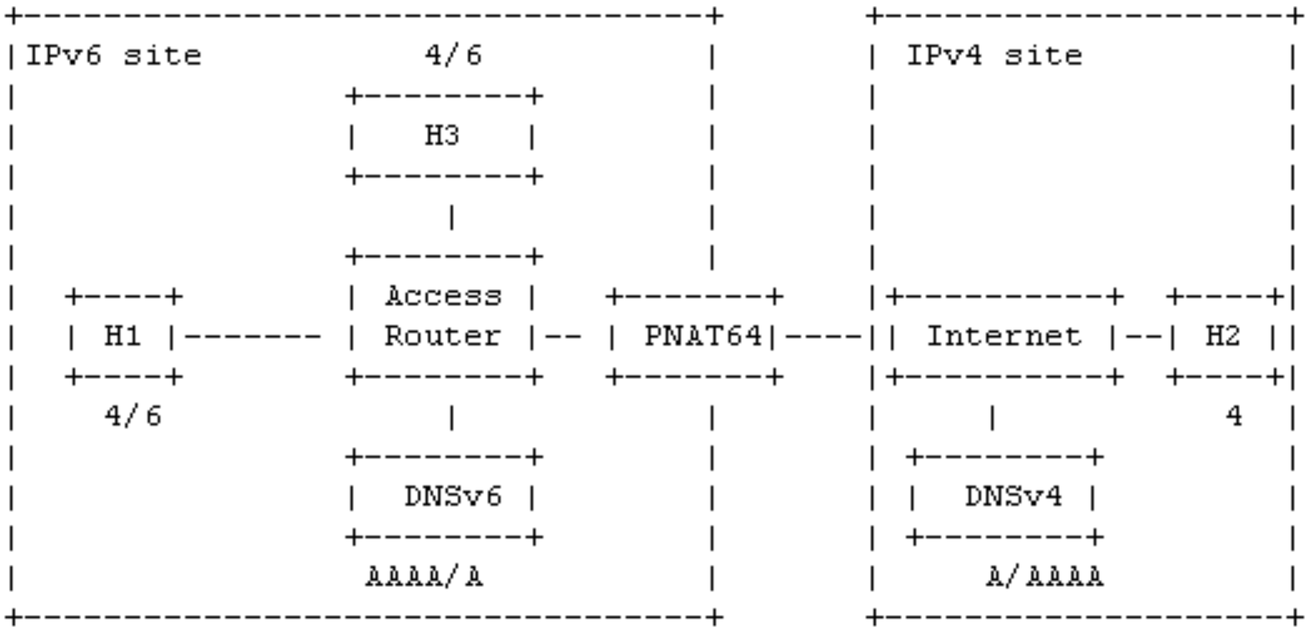
4-4 and 6-4

A	IPv6 only application	Both IPv4 and IPv6 application	Both IPv4 and IPv6 application
B	IPv6 only stack	Dual stack	Dual stack
C	IPv6 only address assignment	Both IPv4 and IPv6 address assignment	Both IPv4 and IPv6 address assignment
D	IPv6 only routing information	Both IPv4 and IPv6 routing information	IPv6 only routing information
Possible solutions	NAT64/IVI64	Dual-stack	Host-based translation

The scenarios we consider are multiple possibilities:

- IPv4-IPv4 application communicate through a IPv6 network;
- IPv4-IPv4 application communicate within a IPv6 network;
- IPv4-IPv6 application communicate
- IPv6-IPv4 application communicate

Network Scenarios



```
+-----------------------------------+      +---------------------+
|IPv6 site           4/6            |      | IPv4 site           |
|                +---------+        |      |                     |
|                |   H3    |        |      |                     |
|                +---------+        |      |                     |
|                     |             |      |                     |
|                +---------+        |      |                     |
|  +-----+       | Access  |  +--------+   |+-----------+ +-----+|
|  | H1  |-------| Router  |--| PNAT64 |---|| Internet  |-| H2  ||
|  +-----+       +---------+  +--------+   |+-----------+ +-----+|
|    4/6              |             |      |      |           4  |
|                +---------+        |      | +---------+         |
|                |  DNSv6  |        |      | |  DNSv4  |         |
|                +---------+        |      | +---------+         |
|                 AAAA/A            |      |     A/AAAA          |
+-----------------------------------+      +---------------------+
```

- H1 and H3 are a dual stack host on which translator is deployed. Both conventional IPv4 application and IPv6 application have been installed
- H2 is IPv4 host reachable in IPv4 Internet
- NAT gateway is used for IPv6-IPv4 translation
- DNSv4 and DNSv6 have both AAAA and A records

H1 and H3 might maintain all potential communications simultaneously

- case 1: IPv4 application in a dual stack host to IPv4 application in a IPv4 host
- case 2: IPv4 application in a dual stack host to IPv6 application in other dual stack host
- case 3: IPv6 application in a dual stack host to IPv4 application in a IPv4 host
- case 4: IPv4 application in a dual stack host to IPv4 application in other dual stack host

Summary

– In order to support numerous legacy IPv4 applications, when the hosts only have an IPv6 connection

– In order to support the compatibility with NAT64

– In order to let the host’s application to understand the IP type of the peer’s application, the host will do according to responded A or AAAA records.

– In order to efficiently use host’s dual stack, such as sending two type of DNS queries (A/AAAA records).

– In order to support v4-v4, v6-v4,v4-v6 communication simultaneously.

Host based IPv6 translation solution: PNAT

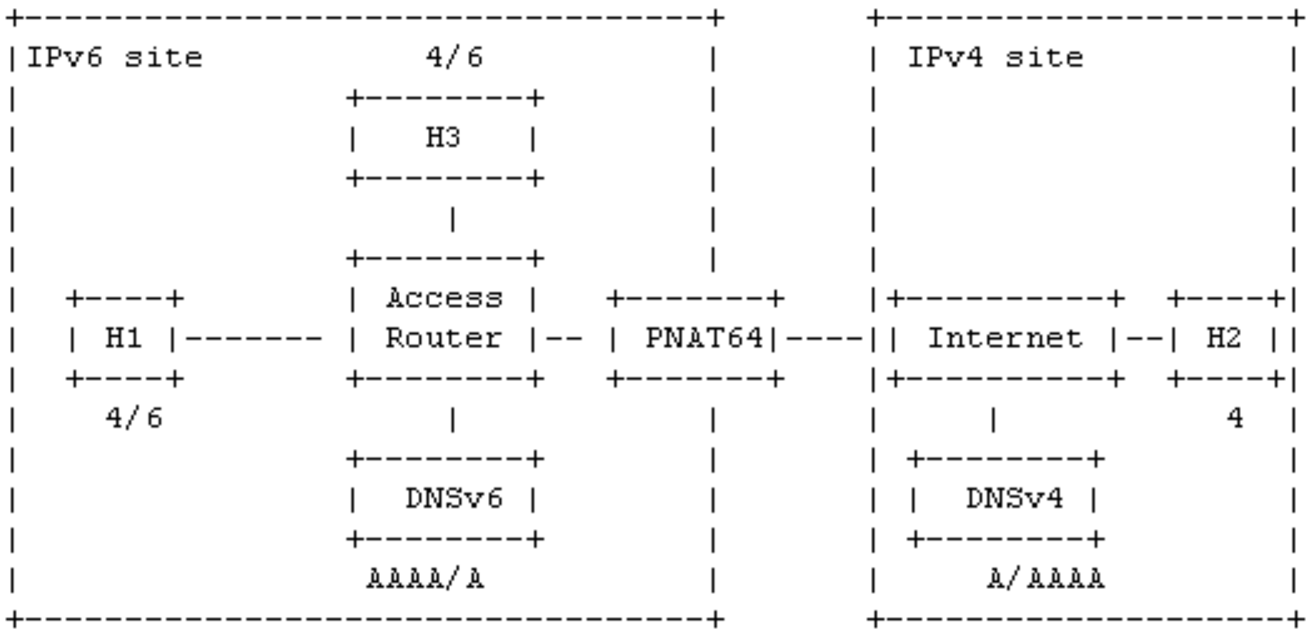
Outline

- How to progress IPv6 transition
- Network scenarios
- PNAT module in the host
- Address assignments and PNAT64 operations
- Signaling procedure of PNAT
- PNAT vs (BIA or BIS)
- PNAT vs Dual-stack lite

How to progress IPv6 transition

- IPv6 deployment could be migrated through network upgrading
  - IPv6 migration is a principally network issue, so upgrading the applications from IPv4 to IPv6 is always supplementary way.
  - There are much more benefits to upgrade the network layer to support IPv6 transition in a host, other than modifying numerous application codes.
  - It's also not easy for application developers to change their experienced network interface API to support dual stack API.
  - IPv6 transition should guarantee conventional IPv4 application continue to communicate with each other in IPv6-only network.
  - IPv6 transition would better support all possible communication scenarios (4-4,4-6,6-4) simultaneously.

Network Scenarios

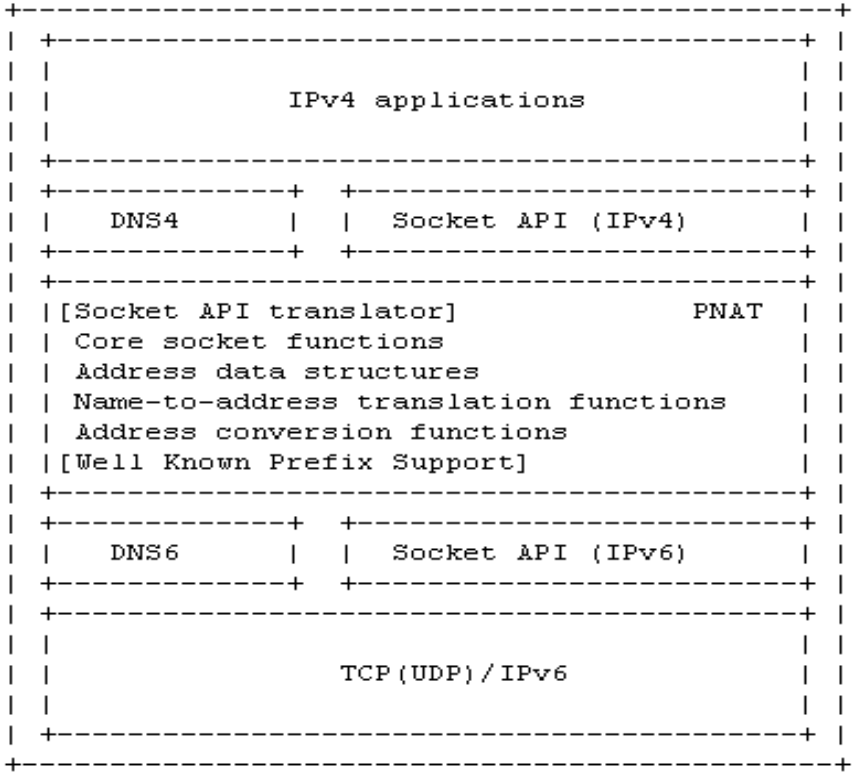


- H1 and H3 are a dual stack host on which translator is deployed
- H2 is IPv4 host reachable in IPv4 Internet
- Access router may allocate the IPv4 and IPv6 prefix to H1 and H2
- PNAT64 gateway is used for IPv6-IPv4 translation and is capable of differentiating the embedded public or private IPv4 address
- DNSv4 and DNSv6 have both AAAA and A records

Scenarios	Descriptions
PNAT44COM	IPv4 application in H1 need to communicate with IPv4 application in H2 or H3
PNAT46COM	IPv4 application in H1 need to communicate with IPv6 application in H3
PNAT64COM	IPv6 application in H1 need to communicate with IPv4 application in H2
PNAT66COM	IPv6 application in H1 need to communicate with IPv6 application in H3

PNAT module in the host

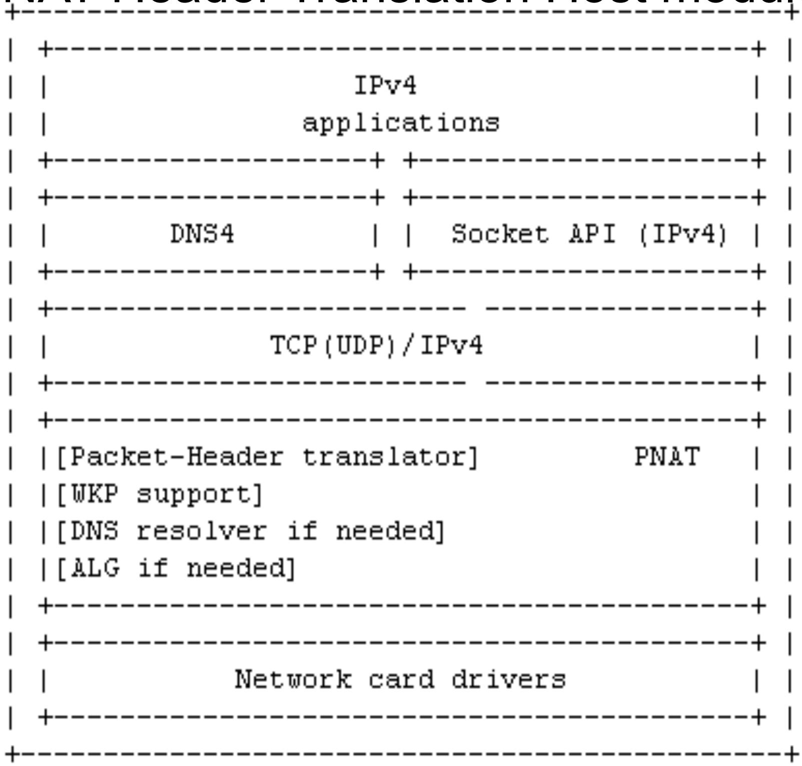
PNAT Socket Translation Host modules



```
+----------------------------------------------+
| +------------------------------------------+ |
| |                                          | |
| |             IPv4 applications            | |
| |                                          | |
| +------------------------------------------+ |
| +-------------+  +-------------------------+ |
| |    DNS4     |  |  Socket API (IPv4)      | |
| +-------------+  +-------------------------+ |
| +------------------------------------------+ |
| |[Socket API translator]          PNAT     | |
| | Core socket functions                    | |
| | Address data structures                  | |
| | Name-to-address translation functions    | |
| | Address conversion functions             | |
| |[Well Known Prefix Support]               | |
| +------------------------------------------+ |
| +-------------+  +-------------------------+ |
| |    DNS6     |  |  Socket API (IPv6)      | |
| +-------------+  +-------------------------+ |
| +------------------------------------------+ |
| |                                          | |
| |             TCP(UDP)/IPv6                | |
| |                                          | |
| +------------------------------------------+ |
+----------------------------------------------+
```

- PNAT inside the host will translate IPv4 socket API into IPv6 socket API
- DNS IPv4 socket call can be translated into IPv6 socket call

PNAT Header Translation Host modules



```
+------------------------------------------+
| +--------------------------------------+ |
| |                IPv4                  | |
| |             applications             | |
| +------------------+ +-----------------+ |
| +------------------+ +-----------------+ |
| |       DNS4       | | Socket API (IPv4) | |
| +------------------+ +-----------------+ |
| +----------------------  --------------+ |
| |             TCP(UDP)/IPv4            | |
| +----------------------  --------------+ |
| +--------------------------------------+ |
| |[Packet-Header translator]     PNAT   | |
| |[WKP support]                         | |
| |[DNS resolver if needed]              | |
| |[ALG if needed]                       | |
| +--------------------------------------+ |
| +--------------------------------------+ |
| |         Network card drivers         | |
| +--------------------------------------+ |
+------------------------------------------+
```

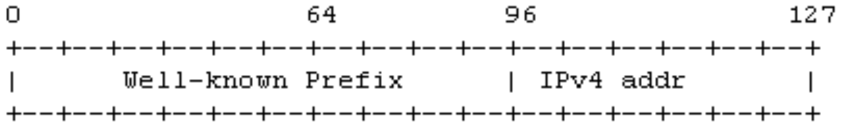
- PNAT inside the host will translate IPv4 header into IPv6 header
- DNS payload will remain original through the header translation

LIR prefix will be used for PNAT source address translation
Well-know prefix will be used for PNAT destination address translation

PNAT Address translation and PNAT64 operations

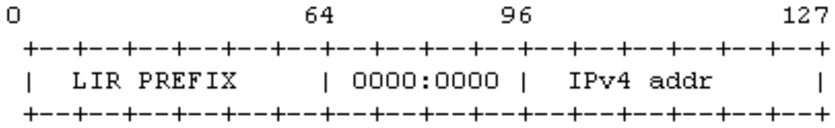
PNAT address translation

- For the destination address



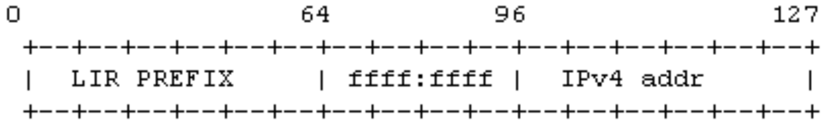
```
0                  64          96                127
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      Well-known Prefix       | IPv4 addr         |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- For the source address, all zero in 65-96 bits is to identify the case of private IPv4 address embedded



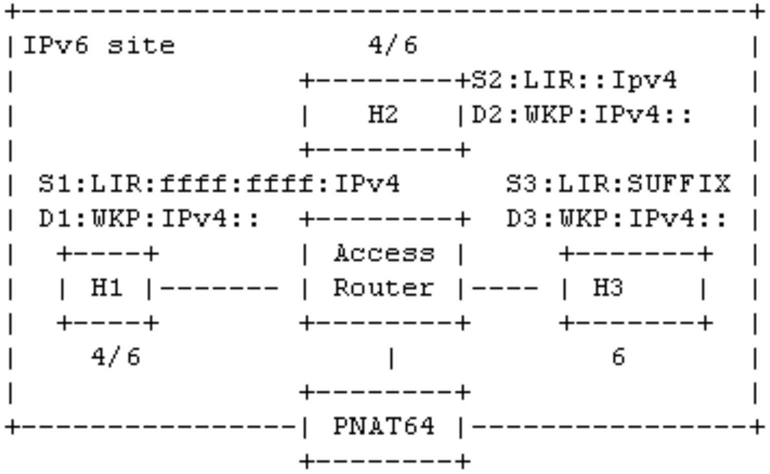
```
0                 64          96               127
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |  LIR PREFIX     | 0000:0000 |  IPv4 addr      |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

- For the source address, all one in 65-96 bits is to identify the case of public IPv4 address embedded



```
0                 64          96               127
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
 |  LIR PREFIX     | ffff:ffff |  IPv4 addr      |
 +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

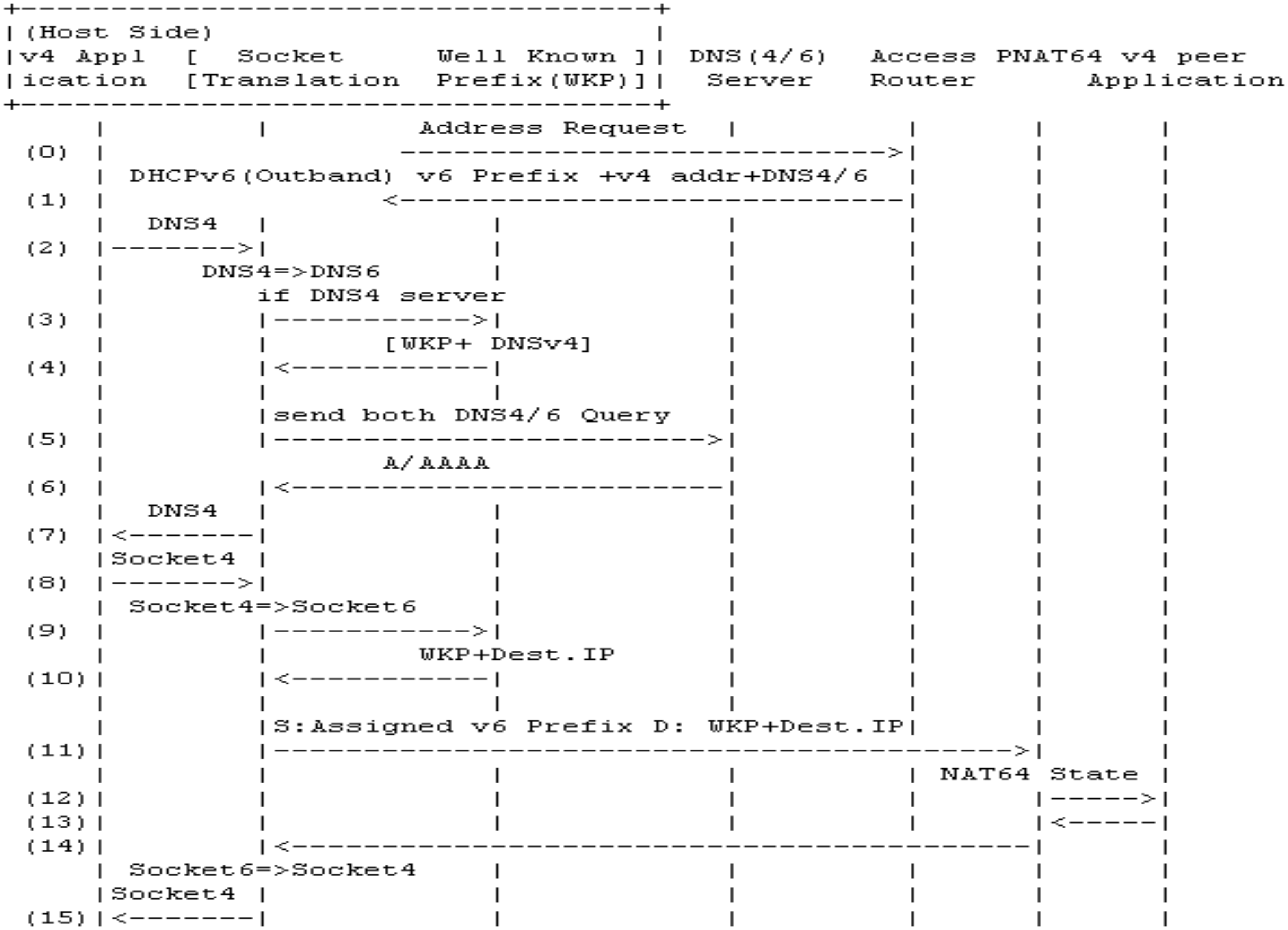
PNAT64 operations



```
+-------------------------------------------------+
|IPv6 site              4/6                       |
|                   +---------+S2:LIR::Ipv4       |
|                   |   H2    |D2:WKP:IPv4::      |
|                   +---------+                   |
| S1:LIR:ffff:ffff:IPv4              S3:LIR:SUFFIX |
| D1:WKP:IPv4::     +---------+     D3:WKP:IPv4:: |
|  +-----+          | Access  |      +--------+   |
|  | H1  |--------  | Router  |----  | H3     |   |
|  +-----+          +---------+      +--------+   |
|    4/6                 |               6        |
|                   +---------+                   |
+-------------------| PNAT64  |-------------------+
                    +---------+
```

Destination addr	Actions
WKP::	perform a translation operation
Source addr	**Actions**
Padding all one in 65-96 bits	Get rid of prefix, record the relationship between IPv4 address and IPv6 prefix
Padding all zero in 65-96 bits	A normal NAT64 procedure
Normal IPv6 address	A normal NAT64 procedure

Signaling procedure of PNAT



```
+-----------------------------------+
|(Host Side)                        |
|v4 Appl  [  Socket     Well Known ]| DNS(4/6)  Access PNAT64 v4 peer
|ication  [Translation  Prefix(WKP)]|  Server   Router       Application
+-----------------------------------+
     |        |       Address Request |         |      |      |
 (0) |               -------------------------->|      |      |
     | DHCPv6(Outband) v6 Prefix +v4 addr+DNS4/6|      |      |
 (1) |              <---------------------------|      |      |
     |  DNS4  |           |           |         |      |      |
 (2) |------->|           |           |         |      |      |
     |    DNS4=>DNS6      |           |         |      |      |
     |       if DNS4 server           |         |      |      |
 (3) |        |---------->|           |         |      |      |
     |        |     [WKP+ DNSv4]      |         |      |      |
 (4) |        |<----------|           |         |      |      |
     |        |           |           |         |      |      |
     |        |send both DNS4/6 Query |         |      |      |
 (5) |        |---------------------->|         |      |      |
     |              A/AAAA            |         |      |      |
 (6) |        |<----------------------|         |      |      |
     |  DNS4  |           |           |         |      |      |
 (7) |<-------|           |           |         |      |      |
     |Socket4 |           |           |         |      |      |
 (8) |------->|           |           |         |      |      |
     | Socket4=>Socket6   |           |         |      |      |
 (9) |        |---------->|           |         |      |      |
     |        |       WKP+Dest.IP     |         |      |      |
 (10)|        |<----------|           |         |      |      |
     |        |           |           |         |      |      |
     |        |S:Assigned v6 Prefix D: WKP+Dest.IP|    |      |
 (11)|        |--------------------------------------->|      |
     |        |           |           |         | NAT64 State |
 (12)|        |           |           |         |      |----->|
 (13)|        |           |           |         |      |<-----|
 (14)|        |<---------------------------------------|      |
     | Socket6=>Socket4   |           |         |      |      |
     |Socket4 |           |           |         |      |      |
 (15)|<-------|           |           |         |      |      |
```

PNAT vs (BIA or BIS)

- The difference:
  - There are no demands to retain mapping table in PNAT44COM, but BIA/BIS still needs
  - PNAT described in detail how it work together with PNAT64, but BIA/BIS doesn't.
  - PNAT host and PNAT64 will process differently for public and private IPv4 source address, but BIA/BIS couldn’t.
  - PNAT can identify peer application type (4 or 6) by responded A or AAAA records, so knows whether the host need to do ALG or not which could avoid NAT464 issue.

PNAT vs Dual-stack Lite

- The difference:
  - PNAT do double translation both in host and PNAT64, and DS-Lite do host tunneling and CGN tunneling plus address translation.
  - PNAT could support multiple scenarios (4-6, 6-4, 4-4), and DS-Lite is targeting 4-4 only; PNAT64 is compatible with NAT64.
  - PNAT host and PNAT64 will process differently for public and private IPv4 source address, and DS-Lite doesn’t.
  - PNAT send two DNS queries with both a and AAAA records.