Internet Engineering Task Force
INTERNET-DRAFT
June, 3, 2002
Expires December, 4, 2002

Alain Durand
SUN Microsystems,inc.

NAT64 - NAT46

<draft-durand-ngtrans-nat64-nat46-00.txt>

Status of this memo

Abstract

This document defines two scalable NAT mechanisms, NAT64 to enable IPv6-only devices to initiate communications with unmodified IPv4 only devices and NAT46 to enable IPv4-only devices to initiate communications with IPv6-only devices.

0. Terminology

In this document, an IPv6-only node refers to a node that implement IPv6 and is configured with an IPv6 address. It may or may not implement IPv4 in the kernel, but if it does, the IPv4 stack is unconfigured.

Similarly, an IPv4-only node may or may not implement IPv6, and if it does, its IPv6 stack is unconfigured.

1. Balkanization of the Internet.

Very early in the design of IPv6, it was decided that there would be no D-day in the transition from IPv4 to IPv6. Such an approach had been experimented in the transition from NCP to TCP when the size of the Internet was only a few hundred hosts and it turned out to be rather chaotic. Even without a D-day, there will be a time where IPv6-only devices will appear in significant number. It is highly probable that, when this will happen, there would still be a very large population of unmodified IPv4-only devices in the Internet, so the question of how much communication is needed in between those two kind of devices is essential to understand when designing transition tools and scenarios. At one end of the spectrum, some say that the value of the Internet is that any device can talk to any other device, thus communication between IPv6-only nodes and IPv4-only nodes should be as seamless as possible. On the other end of the spectrum, some say that new IPv6-only devices will only need services from IPv6 nodes and will not talk to IPv4 nodes, thus the need for communication is minimal and can be handled by application proxies when and if needed. The issue with this later approach is that, even if the final communication is taking place between two IPv6 nodes, it may be the case that the communication set-up process will require third parties, like DNS resolution, LDAP queries, mail MX relay discovery,... and some of the steps in this resolution processes may include sending packet to an infrastructure service only available via IPv4. If IPv6 nodes cannot seamlessly uses those services from IPv4 land, communications between two IPv6-only nodes may or may not be possible. This could lead to the balkanization of the Internet where things may work in some places, but not in others, and troubleshooting will be difficult.

2. Problem statement.

As the number of services that an IPv6 only node may require from the IPv4 Internet is potentially large, it will make sense to try to find a solution at the IP layer that will work for all services.

This memo is proposing a scalable layer 3 solution to enable communication initiated by any IPv6-only node to any unmodified IPv4-only node or the other way round with minimum configuration in the network and very minimum configuration (zero if possible) on the end nodes and without introducing any new security problems.

3. NAT-PT

Issues regarding NAT-PT have been described in [NAT-PTissues].

4. NAT64

This section describe an address translation mechanism to enable an IPv6-only node to initiate a connection to an unmodified IPv4 node.

4.1 Background: IPv4 connections from a dual-stack node

On a dual stack node, when an IPv6 application wants to send a packet to an IPv4 address x.y.z.t, it can encode it in a IPv4-mapped address format ::FFFF:x.y.z.t and let the kernel choose the IPv4 stack. In some sense, this is a form of address translation done within the node.

```
     ::FFFF:x.y.z.t
    ------------|----
   |IPv6 Applica|tion|
   |------------|----|
   |  IPv6  | IP|v4  |
    ------------|----
            |   |
            |   \----------------->
   ======================================> LAN
```

NAT64 proposes to extend this mechanism to IPv6-only nodes.

4.2 Mapping IPv4 address to IPv4-mapped addresses

On IPv6 only nodes implementing [RFC2553], the resolver library may add special code to the getnameinfo() routine to automatically map IPv4 addresses obtained via A record to IPv4-mapped addresses. This way, IPv6 applications using getnameinfo() will always be returned IPv6 addresses.

Application performing direct DNS query for A records should perform this mapping themselves.

4.3 Sending an IPv4-mapped packet

When an IPv6-only node wants to communicate with an IPv4 node, it can send packets to the corresponding IPv4-mapped address on a link using an IPv6 address of the same scope as source address. The IPv6 node should consult its routing table to find out the correct outgoing interface.

```
     ::FFFF:x.y.z.t
    ---|-------------
```

```
      |IPv|6 Application|
      |---|-------------|
      | IP|v6  |        |
       ---|-------------
          |    |
          \----|--------------------->
      ======================================> LAN
```

4.4 Intercepting the IPv4-mapped packets

A NAT64 box can simply inject the ::FFFF:0/96 route in the routing domain where it is willing to perform IPv6 to IPv4 address translation.

```
        ::FFFF:x.y.z.t
       ---|-------------
      |IPv|6 Application|
      |---|-------------|
      | IP|v6  |        |                              NAT64
       ---|-------------                       <----advertising
          |    |                                    ::FFFF:0/96
          \----|----->    --------------             ---------
      ===================|default router|===========|IPv6|IPv4|=======>
                          --------------             ---------
```

4.5 Address translation

IPv6 to IPv4 address translation is then performed in a similar way as described in NAT [RFC3022] and NAT-PT [RFC2766].

4.6 Scalability

NAT64 can be made to scale almost infinitely by adding NAT64 boxes. Each NAT64 will advertise a slightly longer prefix than ::FFFF:0/96. In the example bellow, four NAT64 advertise each 1/4 of the IPv4 address space.

```
                                                      NAT64-0
        ::FFFF:x.y.z.t                             ::FFFF:0000/98
       ---|-------------                             ---------
      |IPv|6 Application|                       ====|IPv6|IPv4|=======>
      |---|-------------|                      |     ---------
      | IP|v6  |        |                      |
       -----------------                       |      NAT64-1
          |    |                               |   ::FFFF:4000/98
          \----|----->    --------------       |     ---------
      ===================|default router|===========|IPv6|IPv4|=======>
                          --------------       |     ---------
                                               |
                                               |      NAT64-2
                                               |   ::FFFF:8000/98
                                               |     ---------
                                               |====|IPv6|IPv4|=======>
                                               |     ---------
                                               |
                                               |      NAT64-3
                                               |   ::FFFF:C000/98
                                               |     ---------
                                                ====|IPv6|IPv4|=======>
                                                     ---------
```

Prefixes advertised by the NAT64 do not have to be all of the same length. For instance, if the network administrator knows there is a lot of traffic for a particular IPv4 prefix, he may want to dedicate a NAT64 for it and let the rest of the traffic go to the main NAT64 box. In the example bellow, the NAT64-0 box is dedicated to translate the traffic for 11.14.14.15

```
                                                      NAT64-0
        ::FFFF:x.y.z.t                             ::FFFF:BEEF/128
       ---|-------------                             ---------
      |IPv|6 Application|                       ====|IPv6|IPv4|=======>
      |---|-------------|                      |     ---------
      | IP|v6  |        |                      |
       -----------------                       |      NAT64-1
          |    |                               |   ::FFFF:0000/96
          \----|----->    --------------       |     ---------
      ===================|default router|===========|IPv6|IPv4|=======>
                          --------------             ---------
```

4.7 Limits of the model

The underlying assumption of this model is that the routes to the NAT64 boxes are stable for the duration of the communication to be translated.

All the traditional limitation of NAT [RFC2993] in IPv4 space also applies here.

5. NAT46

This section describe an address translation mechanism to enable an unmodified IPv4-only node to initiate a connection to an IPv6-only node.

5.1 Background

NAT46 is designed to work for unmodified IPv4-only node, so doing address mapping in the resolver library is out of scope. Some form of ALG has to take place elsewhere in the DNS. The main issue of NAT-PT in this mode of operation is that the DNS ALG is very sensitive to denial of service attacks, as the pool of IPv4 addresses could very easily be exhausted by external attackers. The idea here is to perform the mappings "close" to the originating IPv4 nodes (i.e. in the DNS resolver) instead of "close" to the destination IPv6 node (i.e. in the DNS server), leaving it to the local IPv4 network administrators to offer the service to "bridge" toward the global IPv6 Internet.

5.2 Address mapping

The first thing an unmodified IPv4-only node needs in order to establish a connection to an IPv6-only node is a DNS 'A' record that contains a mapping to the destination IPv6 address. As the underlying assumption is that the IPv4 node cannot be modified, this mapping can only be done by a DNS resolver. So the first step of this process if for the unmodified IPv4-only node to send a A query for a FQDN to a local DNS resolver.

```
       -----------------------------------
      |                                   |
      |                                   |
      |  ----                          --------
      | |IPv4|  'A' query             |  DNS   |
      | |node|----------------------->|resolver|
      |  ----                          --------
      |                                   |
      |            IPv4 network           |
       -----------------------------------
```

Receiving the A query, the local DNS resolver tries to resolve it.

If it succeeds, then the FQDN has one or more IPv4 address already assigned to it and there is no need for translation, the resolver just passes the A records back to the IPv4-only node.

```
       -----------------------------------
      |                                   |
      |                                   |
      |  ----                          --------  'A' query
      | |IPv4|         'A' replies    |  DNS   |--------->
      | |node|<-----------------------|resolver|<--------
      |  ----                          --------  'A' replies
      |                                   |
      |            IPv4 network           |
       -----------------------------------
```

If the A resolution fails, the resolver tries a AAAA query for the same FQDN. If that query fails, there is no IPv6 address assigned to that name, and the resolver returns a failure notification to the IPv4-only node.

```
 -----------------------------------
|                                   |
|                                   |
|  ----                          --------  'AAAA' query fails
| |IPv4|     error notification |  DNS   |--------->
| |node|<-----------------------|resolver|
|  ----                          --------
|                                   |
|            IPv4 network           |
 -----------------------------------
```

If the AAAA query succeeds, the resolver creates mappings between the returned IPv6 address and the same number of IPv4 addresses taken out of a reserved prefix P. IANA considerations for this prefix are covered in Section 7. The resolver then creates A records for those IPv4 addresses, using the TTL includes in the AAAA answers, and return those A records back to the IPv4-only node. The mappings should be expired before the associated TTL.

```
 -----------------------------------
|                                   |
|                                   |
|  ----                          --------  'AAAA' query
| |IPv4|         'A' replies    |  DNS   |--------->
| |node|<-----------------------|resolver|<--------
|  ----                         |mapping | 'AAAA' replies
|                                --------
|                                   |
|            IPv4 network           |
 -----------------------------------
```

5.3 Address translation

The proposition here is to collocate the local DNS resolver with a NAT46 box that will advertise the prefix P to the local network and intercept packets which destination addresses are included in P.

The NAT46 is connected to the IPv6 Internet and is allocated a prefix Q/64 taken out of the IPv6 address block that has been allocated to the network. The NAT46 advertise the prefix Q/64 in the local IPv6 routing domain. Address mapping is then performed:

- IPv4 src x.y.z.t is translated to IPv6 src Q::x.y.z.t
- IPv4 dst m.n.o.p (from prefix P) is translated to the IPv6

addresses that was mapped by the DNS resolver.

The rest of the translation is performed as described in NAT [RFC3022] and NAT-PT [RFC2766].

```
       -----------------------------------
      |                                   |
      |                   route for P  --------
      |  ----               <---------| NAT46  |
      | |IPv4|                        |  DNS   |================>
      | |node|----------------------->|resolver|  IPv6 network
      |  ----  IPv4 packet sent        --------
      |        to the 'mapped'            |
      |        address m.n.o.p            |
      |                                   |
      |            IPv4 network           |
       -----------------------------------
```

When the packet returns from the IPv6 network, as the NAT46 box advertise the prefix Q/64, it can intercept it again and perform the opposite translation.

```
       -----------------------------------
      |                                   |     route for Q/64
      |                                --------  -------->
      |  ----                         | NAT46  |
      | |IPv4|                        |  DNS   |<================
      | |node|<-----------------------|resolver|  IPv6 network
      |  ----                          --------
      |                                   |
      |            IPv4 network           |
       -----------------------------------
```

5.4 Mapping duration

As said in 5.2, mapping should not last more than the TTL associated with the AAAA records. However, after a period of inactivity, the NAT46 box may decide to flush this mapping.

5.5 Scalability

Scalability can be achieved in a similar way as in NAT64 by adding several NAT46 boxes and having each of them advertise a sub-prefix of P. It is possible to partition the IPv4 clients to use different DNS resolver and thus maintain the collocation of DNS resolver and NAT46. Another alternative could be to design a signaling protocol between the DNS resolver and the NAT46 boxes to trigger mapping on/mapping off information. Such a protocol is clearly beyond the scope of this document and is let for future studies.

5.6 Limits of the model

The underlying assumption of this model is that the routes to the NAT46 boxes are stable for the duration of the communication to be translated.

All the traditional limitation of NAT [RFC2993] in IPv4 space also applies here.

Although NAT46 can be used in conjunction with [RFC1918] private address space, it will works for communication started from IPv4 private addresses to global IPv6 addresses. Neither NAT64 nor NAT46 enable an IPv6 node to initiate communications with an IPv4 node that is behind a NAT box and is using [RFC1918] IPv4 private addresses.

6. Security considerations

NAT64 and NAT46 share the same security considerations as IPv4 NAT as they operate the same way. The security consideration raised in [NAT-PTissues] do not apply to NAT64. If such an attack is launched against NAT46, the effect will be limited to the IPv4 site covered by NAT46 and will not prevent communication from other nodes in the IPv4-only Internet to initiate communication with IPv6 only nodes.

7. IANA Considerations

As NAT46 can be used in conjunction of [RFC1918] addresses, the prefixes 10/8, 172.16/12 and 192.168/16 cannot be use for address mapping.

Allocating a /8 prefix would enable 2^24 (about 16 millions) contexts. Allocating a /16 prefix would enable 2^16 (about 65 thousands) contexts. Allocating a /24 prefix would enable 2^8 (256) contexts. Decision on the prefix length to allocate will be done in the NGtrans mailing list.

8. Author addresses

Alain Durand
SUN Microsystems, Inc
901 San Antonio Road MPK17-202
Palo Alto, CA 94303-4900

USA
Mail: Alain.Durand@sun.com

9. References

[RFC1918] Address Allocation for Private Internets. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. February 1996.

[RFC2553] Basic Socket Interface Extensions for IPv6. R. Gilligan, S. Thomson, J. Bound, W. Stevens. March 1999.

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

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

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

[NAT-PTissues] Issues with NAT-PT DNS ALG in RFC2766, A. Durand, draft-durand-natpt-dns-alg-issues-00.txt, work in progress.

10. Full Copyright Statement

THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.