NAT64—Stateless versus Stateful

Last updated: July 2011

Within the IPv4 networked world Network Address and Port Translation (NAPT, but also called NAT) between a public IPv4 address and a private IPv4 address was created around 2001 to address the ongoing depletion of the public IPv4 address pool. Another solution to resolve address depletion was a technology identified as VLSM (variable length subnet Masks) and served as a good short term solution. The long term solution for IPv4 address exhaust resulted in a forklift IP upgrade from IPv4 towards IPv6.

This paper will discuss briefly traditional IPv4 address translation, followed with some elements of cross IPv4 and IPv6 NAT translation solutions, including the historical NAT-PT solution.

What is in this paper:

- IPv4-IPv4 Address translation—NAPT (also known as NAT44)
- NAT-PT—(IPv6 Network Address Translation—Protocol Translation)
- IPv6 to IPv4 Network Address Translation
  - Stateless NAT64 translation
  - Statefull NAT64 translation

IPv4 Network Address and Port Translation (IPv4 NAT and NAPT)

The IPv4 network address and port translation (NAPT, RFC3022) we all know is mainly based upon private IPv4 addresses space towards public IPv4 address space. This is utilized most frequently at residential Home-Gateways during last decade and is a stateful technology. It is stateful because each translation creates state within the NAPT device, necessary for the return packet to be routed. The initial purpose of NAPT was to multiplex many users in a single IPv4 address on the global Internet. The example shown here is with Cisco NAT (with overload technology) where many devices using private RFC1918 addresses on the inside of the NAT44 can use a single global unique IPv4 address to communicate on the global IPv4 Internet. The Cisco Command Line Interface (CLI) shown below gives an indication of the stateful mapping between the inside and outside IP addresses and port numbers used.

Table 1. Stateful Mapping Between the Inside and Outside IP Addresses and Port Numbers

```
Router# show ip nat translations

Pro Inside global        Inside local       Outside local      Outside global
udp 172.16.233.209:1220  192.168.1.95:1220  172.16.2.132:53      172.16.2.132:53
tcp 172.16.233.209:11012 192.168.1.89:11012 172.16.1.220:23    172.16.1.220:23
tcp 172.16.233.209:1067  192.168.1.95:1067  172.16.1.161:23    172.16.1.161:23
```

The usage of an IPv4 Network Address Translator device goes from the assumption that the L4 protocol used is either UDP or TCP, and that the two communicating end-stations support IPv4 as protocol.

An important issue of NAPT isn't so much on "what amount of memory do you need" or "what is the lookup time in the implied data structure". The issue is the effect on applications. As a result of NAPT, we can't put publicly-accessible applications behind the firewall, and have to think about their security separately.

NAT-PT–(IPv6 Network Address Translation–Protocol Translation)

The first solution the IETF (Internet Engineering Task Force—www.ietf.org) developed to allow IPv4 'ONLY' hosts to speak to IPv6 'ONLY' hosts and vice versa is known as NAT-PT (RFC2766). This first translation solution between IPv4 and IPv6 existed out of mainly two parts:

- Domain Name System (DNS) Application Level Gateway (ALG)
  - When a IPv6 user does a DNS name query for a device with only IPv4 connectivity (ie. www.foo4.com), then the DNS request must be translated from being a IPv6 DNS request towards a IPv4 DNS request. This is a function achieved by the DNS ALG embedded within the NAT-PT technology.
  - Similar when a IPv4 only user does a DNS name query for a device with only IPv6 connectivity the DNS ALG embedded in the NAT-PT translates this DNS request
- Address Family Translation (AFT)
  - When an IPv6-only device wants to send an IPv6 data packet to an IPv4-only device then, there is need for a translation of the IPv6 protocol header into a IPv4 protocol header so that the recipient device can understand the data. On the return path after the IPv4-only device sends a response packet, a translation is needed from IPv4 towards IPv6, so that the IPv6-only device can understand the response packet.
  - The foundational algorithm of the IPv4/IPv6 protocol translation used within NAT-PT is based upon a technology called "Stateless IP/ICMP Translator" also known as SIIT (RFC2765). This same technology describes also the IPv6-initiated IPv6-to-IPv4 protocol translation known as stateless NAT64.

Over the years NAT-PT usage has proven as technology to be too complex to maintain scalable translational services, resulting in it being declared historical (RFC4966—Reasons to Move the Network Address Translator—Protocol Translator (NAT-PT) to Historic Status). However, translational technology has been made more modular and segmented, and replacements for translations between IPv6 and IPv4 have been created within a focussed translation framework by the IETF in the Behave working group (http://datatracker.ietf.org/wg/behave/charter/).

Stateless NAT64—Stateless translation between IPv4 and IPv6

RFC6145 (IP/ICMP Translation Algorithm) replaces RFC2765 (Stateless IP/ICMP Translation Algorithm (SIIT)) and provides a stateless mechanism to translate a IPv4 header into an IPv6 header and vice versa. Due to the stateless character this mechanism is very effective and highly fail safe because more as a single—or multiple translators in parallel can be deployed and work all in parallel without a need to synchronize between the translation devices.

The key to the stateless translation is in the fact that the IPv4 address is directly embedded in the IPv6 address. A limitation of stateless NAT64 translation is that it directly translates only the IPv4 options that have direct IPv6 counterparts, and that it does not translate any IPv6 extension headers beyond the fragmentation extension header; however, these limitations are not significant in practice.

With a stateless NAT64, a specific IPv6 address range will represent IPv4 systems within the IPv6 world. This range needs to be manually configured on the translation device. Within the IPv4 world all the IPv6 systems have directly correlated IPv4 addresses that can be algorithmically mapped to a subset of the service provider's IPv4

addresses. By means of this direct mapping algorithm there is no need to keep state for any translation slot between IPv4 and IPv6. This mapping algorithm requires the IPv6 hosts be assigned specific IPv6 addresses, using manual configuration or DHCPv6.

Stateless NAT64 will work very successful as proven in some of the largest networks, however it suffers from some an important side-effect: Stateless NAT64 translation will give an IPv6-only host access to the IPv4 world and vice versa, however it consumes an IPv4 address for each IPv6-only device that desires translation -- exactly the same as a dual-stack deployment. Consequentially, stateless NAT64 is no solution to address the ongoing IPv4 address depletion. Stateless NAT64 is a good tool to provide Internet servers with an accessible IP address for both IPv4 and IPv6 on the global Internet. To aggregate many IPv6 users into a single IPv4 address, stateful NAT64 is required.

NAT64 are usually deployed in conjunction with a DNS64. This functions similar to, but different than, DNS-ALG that was part of NAT-PT. DNS64 is not an ALG; instead, packets are sent directly to and received from the DNS64's IP address. DNS64 can also work with DNSSEC (whereas DNS-ALG could not).

Stateful NAT64—Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers

Stateful NAT64 multiplexes many IPv6 devices into a single IPv4 address. It can be assumed that this technology will be used mainly where IPv6-only networks and clients (ie. Mobile handsets, IPv6 only wireless, etc…) need access to the IPv4 internet and its services.

The big difference with stateful NAT64 is the elimination of the algorithmic binding between the IPv6 address and the IPv4 address. In exchange, state is created in the NAT64 device for every flow. Additionally, NAT64 only supports IPv6-initiated flows. Unlike stateless NAT64, stateful NAT64 does ‘not’ consume a single IPv4 address for each IPv6 device that wants to communicate to the IPv4 Internet. More practically this means that many IPv6-only users consume only single IPv4 address in similar manner as IPv4-to-IPv4 network address and port translation works. This works very well if the connectivity request is initiated from the IPv6 towards the IPv4 Internet. If an IPv4-only device wants to speak to an IPv6-only server for example, manual configuration of the translation slot will be required, making this mechanism less attractive to provide IPv6 services towards the IPv4 Internet.

DNS64 is usually also necessary with a stateful NAT64, and works the same with both stateless and stateful NAT64.

Summary

While stateful and stateless NAT64 perform the task of translating IPv4 packets into IPv6 packets and vice versa, there are important differences as explained in previous sections. The following table provides a high-level overview of the most relevant differences.

Table 2. Differences Between Stateless NAT64 and Stateful NAT64

Stateless NAT64	Stateful NAT64
1:1 translation	1:N translation
No conservation of IPv4 address	Conserves IPv4 address
Assures end-to-end address transparency and scalability	Uses address overloading, hence lacks in end-to-end address transparency
No state or bindings created on the translation	State or bindings are created on every unique translation
Requires IPv4-translatable IPv6 addresses assignment (mandatory requirement)	No requirement on the nature of IPv6 address assignment

Requires either manual or DHCPv6 based address assignment for IPv6 hosts	Free to choose any mode of IPv6 address assignment viz. Manual, DHCPv6, SLAAC

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-676277-00 07/11