Volume : 2 | Issue : 6 | June 2013 • ISSN No 2277 - 8179

Research Paper

Computer Science

KEYWORDS :

Present State of IPv6 NAT

Sridevi Assistant Professor, Dept.of Computer Science, Karnatak University, Dharwad.

ABSTRACT

This research paper will indicate the current situation of NAT for IPv6. It will also illustrate the current approaches, and discuss the techniques and the security issues of IPv6 NAT.

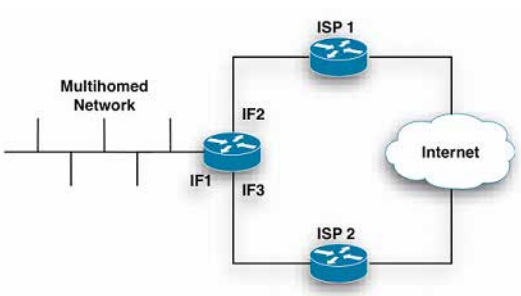
1. Introduction

On 3rd February 2011 the Internet Assigned Numbers Authority (IANA) assigned the remaining last five /8s of IPv4 address space to the Regional Internet Registries (RIR). The IPv4 X-Day arrived. Nowadays, the switch to IPv6 is still in progress and the day is approaching, where the last IPv4 address will be allocated. Although IPv6 is standardised by the Internet Engineering Task Force (IETF) since 1998, the standardisation process is not concluded yet and there are still open questions. For instance, a decision about Network Address Translation (NAT) for IPv6 was first made in June 2011 with the Internet RFC 6296. The corresponding document is still marked as experimental draft. There are divergent opinions on IPv6 NAT amongst the IETF experts. On the one hand, NAT as flow multiplexer should be unnecessary with IPv6, as IPv6 comes with a considerable IP address space to assign an IPv6 address for every electronic device worldwide. These issues would be solved with a NAT-free IPv6. On the other hand, NAT is demanded by enterprises for multihoming, which will still be requested in IPv6 networks. The late decision on IPv6 NAT and the long period of uncertainty caused that only a few implementations are available at the present day. Solutions for enterprise networks and implementations in the public domain are rarely found.

2. Multihoming for IPv4

With multihoming, a site or client is connected to the Internet through more than one ISP. If one connection fails, the router will change the upstream network to one of the operational connections. Multihoming provides fault tolerance and guarantees a reliable connectivity to the Internet. These two characteristics make multihoming for enterprises more and more popular.

Figure 1: Multihomed Network Connected to Two ISPs



Multihoming for IPv4 with Provider Independent (PI) prefix, or Provider Aggregatable (PA) is realised over the Boarder Gateway Protocol (BGP). When using a PI prefix, the multihomed site announces its PI Prefix through all its ISPs. The PI prefix is than stored multiple times in the Internet core routers' routing tables. When using a PA prefix, the PA prefix is never be propagated beyond the provider's Autonomous System (AS). The filtering of the customer's IP prefix is the ISP's responsibility. Only the ISP's PA prefix is sent to External BGP (EBGP) peers[1]. It must be noted that not every enterprise neither is eligible to register nor can afford a PI address space (Hilliard, 2009). In the case of using a Provider Aggregatable (PA) address space, NAT is required to switch the Internet connection without the demand to renumber the internal network each time. With the increasing use of multihoming and the enlarged IPv6 address length, the routing table size might become a serious issue in the future when using PI prefixes. Using PA prefixes avoids the routing and network management issues, but creates new challenges with IPv6 [11].

2.2 IPv6 Address Structure

An IPv6 address has the size of 128 bits. Depending on the addressing and routing methodology, the address is classified as unicast, multicast or anycast address. For each address type, the structure of the IP address slightly differs. In section 3.1, IPv6 NAT is described for the simplest case that a unicast address is translated to a unicast address.

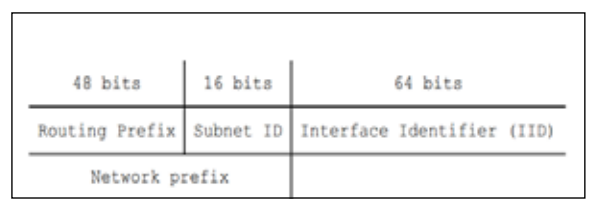


Fig 2: IPv6 Unicast Address Format

The 128 bits are divided into 3 parts. The 64 last significant bits are noted as Interface Identifier. The interface identifier is used to identify the host's network interface. A network interface can be reachable under more than one IPv6 IP address. The first 64 most significant bits are noted as network prefix and are used for routing. The network prefix is divided into the 48 most significant bits noted as routing prefix and the remaining 16 bits noted as subnet id[5].

3. NAT for IPv6

On the one hand, IPv6 has 128 bits long addresses. Compared to the 32 bits long IPv4 addresses, address multiplexing should no longer be required to connect a network over one public IP address with the aim to save IP addresses. Each device can be connected to the Internet with its own public IP address directly. Besides that, there is no need of NAT traversal techniques to solve IPv4 NAT issues that, for instance, occur with Next Generation Network services, which use UDP as stateless transport layer protocol, such as Voice over IP. On the other hand, IPv6 NAT might still be useful in combination with multihoming, server load balancing, failover, or demilitarised zones (DMZs). In June 2011 the IETF released with the Internet RFC 6296 an experimental document about Network Prefix Translation (NPTv6). The authors emphasise at the beginning that they do not recommend the use of any Network Address Translation technology for IPv6[15]. That underlines the experimental classification of the document, which also means that NPTv6 needs not necessary evolve to a standard.

3.1 Network Prefix Translation

The IETF suggests just a stateless 1:1 NAT for IPv6 to avoid the same architectural issues associated with the traditional stateful IPv4-to-IPv4 NAT (NAPT44) techniques. The primary objective of NPTv6 is an address independent translation just as IPv4-to-IPv4 NAT. Address independence avoids renumbering on the client side and on the provider side, in the case that a renumbering event occurs. Additionally, NPTv6 is described as a two-way, checksum-neutral, algorithmic translation function that supports inbound connection establishment and acceler-

Volume : 2 | Issue : 6 | June 2013 • ISSN No 2277 - 8179

ates the translation process compared to stateful NAT. With NPTv6, it is needless to recalculate checksums, to maintain flow states, to map ports, and to rewrite transport layer headers. At first, the NPTv6 Translator ensures that the internal and external prefixes are the same length. If necessary, the shortest prefix is extended with zeroes. The prefixes are then zero-extended to /64 for calculating the one's complement checksum of the 16-bit words of the /64 external prefix and the /64 internal prefix. After that, an "adjustment" is calculated, which is defined as the difference between the previous calculated one's complement checksums (internal minus external). The adjustment is constant for the lifetime of the NPTv6 Translator configuration and is used for translating the routing prefix. The adjustment is subtracted from the 16-bit subnet field on packets traversing from the internal network to the external network. It is added to the 16-bit subnet field for packets traversing the other direction. If the result of the 16-bit subnet field is OxFFFF, it is overwritten as zero.

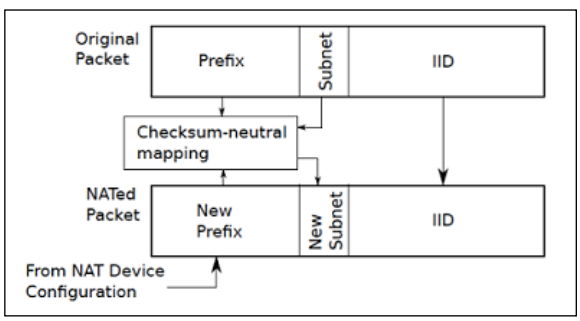


Fig 3: Stateless mapping

The checksum-neutral algorithm works slightly different for prefixes of other length than /48. In that case, the adjustment is added to, or subtracted from other bit-positions of the address. The exact bit-positions are specified in the Internet RFC 6296.

3.2 Short Overview of Various Applications for NPTv6

NPTv6 is implemented in routers and translates addresses between two or more networks. In the simplest case, the internal network uses IPv6 Unique Local Addresses (ULAs) as internal prefix and the external network uses globally routable addresses. Then, the NPTv6 Translator translates the internal prefix to a globally routable prefix on all outgoing packages traversing the router, and vice versa.

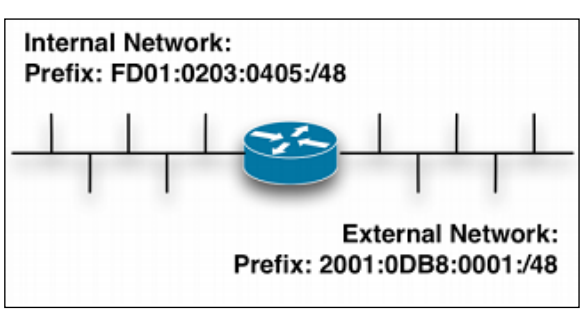


Fig 4: A Simple NPTv6 Translator

The external network must not necessary be a public network. NPTv6 also translates between two private networks, where both networks use ULA prefixes. In that case, the external network can be addressed by the corresponding ULA prefix. It is also possible to address it by a global unicast address, even when the external network uses an ULA prefix internally.

When using more than one NPTv6 Translator, which all are configured with the same internal prefix and the same external prefix, NPTv6 can also be used for load sharing or for failover. For multihoming, the multiple NPTv6 Translators are configured with the same internal but different external prefixes, since they are connected to different external networks. In this scenario, an external system is unable to determine which external address is used[15]. Figure 4 illustrates the translators as two separate routing devices for better visualisation, but in general multiple translators are implemented in one physical routing device.

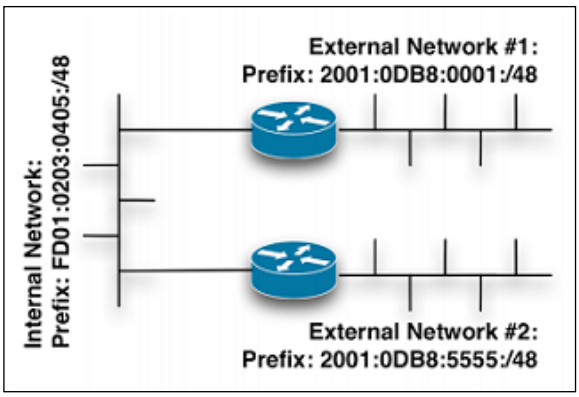


Figure 5: Parallel Translators with Different Upstream NetworkS

3.4 Security Issues

NPTv6 operates as static NAT and exposes all hosts unprotected to the external network. The hosts are vulnerable for attacks, when no firewall is deployed. It might be the case that customers assume that NPTv6 offers the same slight protection just as dynamic NAT. NAT in general does not replace a firewall. Therefore it is important that NPTv6 devices should also implement a firewall functionality, which can be enabled by the customer individually. When calculating the checksum-neutral modification of the subnet identifier, it might be desirable to perform a randomized altering. This makes it more difficult for attackers to guess a valid subnet identifier at an advertised network prefix. Furthermore, interactions of NPTv6 with IKEv2/IPsec NAT traversal are still unexamined in detail at the time of writing [15].

3.5 Legal Aspects

NPTv6 is not just useful for multihomed networks. With IPv6, the anonymity of the Internet will be lost, if ISPs assign static IPv6 addresses to their clients. With a static IPv6 address assignment, users are much easier traceable. Tracking cookies will no longer be needed, since the user can be identified by his or her static IP address. As IP addresses are usually stored in server logs, data retention will easily be deployable. This might cause a violation of the Data Protection Act in some countries. An NPTv6 implementation in access routers can counteract that issue. With NPTv6, the ISPs can assign their clients new IP addresses every 24 hours, without the need of renumbering on the client site. This would also require a dynamic change of the Interface Identifier, which is discussed in the Internet RFC 3041[12].

4. Current Implementations

In view of the fact that the IETF first made its decision about NAT for IPv6 in 2011, there are only a few IPv6 NAT implementations available at the moment. At the time or writing, IPv6 NAT is only integrated in OpenBSD and FreeBSD, as Unix kernel based operating systems, with the pf packet filter application. The lack of a standard emerged various implementations with different NAT strategies. The most known implementation in the public domain is conducted [10], which adopts the IETF constraints for a stateless NAT.

4.1 NF NAT66

Moës designed his implementation before the IETF released the Internet RFC 6296. He implemented both stateless and stateful IPv6 NAT for the netfilter framework, which is embedded into the Linux kernel. He used the current netfilter implementation of IPv4-IPv4 address translation (NAT44) as basis and named his implementation NAT66 regarding to IPv6-IPv6 address translation. Moës deployed the source code for his netfilter modules to SourceForge in Jule 2011. It is assumed that his

implementation achieves the same throughput as the current NAT44 netfilter implementation.

His stateless NAT66 algorithm operates just as the previously described NPTv6 algorithm. Therefore, it avoids recalculation of the checksums and operates transparent for the upper layers. The NAT algorithm manipulates the routing prefix and uses the subnet id to compensate the alteration of the routing prefix. Both changes lead to the same checksum as with the original network prefix. Since the algorithm leaves the interface identifier unaltered, the process is also transparent for the upper layers.

The stateful NAT66 algorithm operates just like NAPT44. The NAT device maintains a translation table with IP addresses and ports. The modification of the IP header is not checksum-neutral so that the IP header checksum and checksums in higher layers have to be recalculated, which causes a higher computational effort. In contrast to other implementations, which just add two targets to Xtables, Moës'

implementation involves connection tracking, is able to deal with ICMP error messages and of course deals with stateful N:1 NAT. Because Moës' NAT66 is developed under Linux and utilises netfilter, it can be cross-compiled to all Linux based systems. Therefore, it is feasible to deploy it to OpenWrt and use it on a wide number of routers different vendors.

4.2 MAP66

MAP66 is another IPv6 NAT realisation in the public domain, implemented by Sven-Ola Tücke. The deployed kernel object covers only static NAT for IPv6 and is inspired by the IETF Internet draft about IPv6-to-IPv6 Network Address Translation[15].

4.3 RAWNAT

RAWNAT is an xtables add-on for stateless NAT. Xtables and the add-ons are available as package for all Linux distributions. There are also packages for RAWNAT on OpenWrt.

4.4 Netfilter

In November 2011, the netfilter team decided to release patches for the netfilter framework that extend netfilter with IPv6 NAT (McHardy, 2011). The supervision by the netfilter team should ensure that the implementation is well tested and nobody has to implement a solution on his own. At the time of writing, the netfilter core team has just released the patches, which must be integrated manually. They are not included into the official packages yet.

4.5 Cisco IOS

Cisco, a multinational networking equipment vendor, only supports Network Address Translation-Protocol Translation (NAT-PT), an IPv6 to IPv4 translation mechanism, in their current Cisco IOS release 15.2. It allows IPv6-only devices to communicate with IPv4-only devices and vice versa (Cisco Systems, 2011 p.792). Cisco does not offer any IPv6-to-IPv6 1:1 NAT or N:1 NAT solution yet.

4.6 Juniper Junos OS

In contrast to Cisco Systems, Juniper Networks, Cisco's competitor, fully supports IPv6-to-IPv6 NAT with the current Junos OS release 12.2[7].

5. Conclusions

The long period of uncertainty about IPv6 NAT emerged an unclear situation for NAT deployments. Today there are only a few implementations available, whereby Moes' solution seems to be the most sophisticated work in the public domain. It offers both stateless and stateful NAT for IPv6 and since the source code is published, it can be cross-compiled to all Linux based systems.

For the enterprise market, only Juniper Networks offers a NPTv6 implementation for their Junos OS. This seems to be contradictory, as NAT for IPv6 was primarily demanded by enterprises to further support multihoming in IPv6 networks. The lack of enterprise solutions can be explained by the lack of NPTv6 as a standard. The poor circulation of NPTv6 is the reason for the absence of reliable statistics about the data throughput of current solutions. Therefore, it is not possible to predict the behaviour of NPTv6 in a practical environment. It must be assumed that NPTv6 comes with a performance advantage compared to NAPT44.

NPTv6, as static NAT, may prevent the dropping of flows traversing NAT Translators. It also reduces the administrational overhead and the effort for deploying services, which had to use NAT Traversal utilities in IPv4 NAT environments. An NPTv6 rollout must be carried out with respecting the security issues of NPTv6. Every NPTv6 deployment should be protected by a firewall. Furthermore, it should be clear that there are still protocols such as IKEv2/IPsec and UNSAF that might break while traversing the NAT Translator. As described, NPTv6 is very scalable, since the Translator function works the same on each device. Therefore, it is possible to use a plurality of NAT Translators. That makes NPTv6 ideal for load sharing and failover. Even when the IETF recommends deploying IPv6 without any NAT, NPTv6 is the IETF's suggested solution for IPv6 NAT. It is uncertain whether NPTv6 will be standardised. As it is not only needed for multihoming, NPTv6 can at least become a standard, as soon as governments define a dynamic address assignment for ISPs as law, to be conform with local Data Protection Acts.

REFERENCE

[1] Abley, J., Black, B. and Gill, V. (2003). Goals for IPv6 Site-Multihoming Architectures. Internet RFC 3582. The Internet Engineering Task Force. | [2] Abley, J. et al (2005). IPv4 Multihoming Practices and Limitations. Internet RFC 4116. The Internet Engineering Task Force. | [3] Cisco Systems. (2011, July). Cisco IOS IPv6 Command Reference. Retrieved September 14, 2012 from Cisco Systems: http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_cr_book.pdf | [4] Hilliard, N. (2009). Contractual Requirements for Provider Independent Resource Holders in the RIPE NCC Service Region. RIPE 452. Ripe Network Coordiantion Centre. | [5] Hinden, R. and Deering, S. (2006). IP Version 6 Addressing Architecture. Internet RFC 3513. The Internet Engineering Task Force. | [6] Hinden, R. and Haberman, B. (2005). Unique Local IPv6 Unicast Addresses. Internet RFC 4193. The Internet Engineering Task Force. | [7] Juniper Networks. (2012, June 06). Network Address Translation. Retrieved September 14, 2012 from Network Security Solutions - Networking Performance Optimization:http://www.juniper.net/techpubs/ | en_US/ junos12.2/information-products/pathway-pages/services-interfaces/router-services-network-address-translation.pdf | | [8] Kaufman, C. et al (2010). Internet Key Exchange Protocol Version 2 (IKEv2). Internet RFC 5998. The Internet Engineering Task Force. | [9] McHardy, P. (2011, November 21). [RFC PATCH 00/17] netfilter: IPv6 NAT. Retrieved September 14, 2012 from Linux Netfilter Devel: http://68.183.106.108/lists/netfilter-devel/msg19979.html | [10] Moës, T. (2011). IPv6 Address Translation in a Linux Kernel. A Thesis Submitted in partial fulfilment of the Requirements of University of Liège for the Degree of Master of Computer Engineer. Liège: University of Liège. | [11] Naderi, H. and Carpenter, B. E. (2011). A Review of IPv6 Multihoming Solutions. ICN 2011 : The Tenth International Conference on Networks. International Academy, Research, and Industry Association. | [12] Narten, T., Draves, R. and Krishnan, S. (2007). Privacy Extensions for Stateless Address Autoconfiguration in IPv6. Internet RFC 3041. The Internet Engineering Task Force. | [13] Srisuresh, P. and Holdrege, M. (1999). IP Network Address Translator (NAT) Terminology and Considerations. Internet RFC 2663. The Internet Engineering Task Force. | [14] Wasserman, M. and Baker, F. (2008). IPv6-to-IPv6 Network Address Translation (NAT66). draft-mrw-behave-nat66-02. The Internet Engineering Task Force. | [15] Wasserman, M. and Baker, F. (2011). IPv6-to-IPv6 Network Prefix Translation. Internet RFC 6296. The Internet Engineering Task Force. |