BEHAVE WG	M. Wasserman	
Internet-Draft	Sandstorm Enterprises	
Expires: May 7, 2009	F. Baker	
	Cisco Systems	
	November 03, 2008	

TOC

IPv6-to-IPv6 Network Address Translation (NAT66)
draft-mrw-behave-nat66-01.txt

Status of this Memo

Abstract

This document describes a stateless, transport-agnostic IPv6-to-IPv6 Network Address Translation (NAT66) function that provides the address independence benefit associated with IPv4-to-IPv4 NAT (NAT44) while minimizing, but not completely eliminating, the problems associated with NAT44.
This document also describes an address mapping option for NAT66 that offers the topology hiding benefit associated with NAT44 at the cost of additional state in the NAT66 device.

Table of Contents

---

1. Requirements Terminology

TOC

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] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

2. Introduction

TOC

This document describes a stateless, transport-agnostic IPv6-to-IPv6 Network Address Translation (NAT66) function that provides the address independence benefit associated with IPv4-to-IPv4 NAT (NAT44) while minimizing, but not completely eliminating, the problems associated with NAT44.

NAT66 does not include a port mapping function, and both of the defined address mapping mechanisms use checksum-neutral algorithms. This avoids the need for a NAT66 device to re-write transport layer headers, making it feasible to deploy new or improved transport layer protocols without upgrading NAT66 devices. Because NAT66 does not involve re-writing transport-layer headers, NAT66 will not interfere with encrypting the full IP payload in many cases.

The default NAT66 address mapping mechanism is purely algorithmic, so NAT66 devices do not need to maintain per-node or per-connection state,

allowing deployment of more robust and adaptive networks than can be deployed using NAT44. Since the default NAT66 mapping can be performed in either direction, it does not interfere with inbound connection establishment, thus allowing internal nodes to participate in direct peer-to-peer applications.
This document also defines an address mapping option for NAT66 that offers the topology hiding benefit associated with NAT44. This mechanism involves the configuration or dynamic maintenance of some per-node state in the NAT66 device. So, when used with this optional address mapping mechanisms, NAT66 will have greater negative impact on direct peer-to-peer applications, and on the robustness and reliability of the network. These trade-offs are discussed later in the document.
Although NAT66 compares favorably to NAT44 in several ways, it does not eliminate all of the architectural problems associated with IPv4 NAT. [RFC2993] (Hain, T., “Architectural Implications of NAT,” November 2000.). NAT66 involves modifying IP headers in transit, so it is not compatible with security mechanisms that involve end-to-end encryption of the IP header or mechanisms, such as AH, that provide integrity protection for the IP header. NAT66 may interfere with the use of application protocols that transmit IP addresses in the application-specific portion of the IP packet. These applications currently require application layer gateways (ALGs) to work correctly through NAT44 devices, and similar ALGs may be required for these applications to work through NAT66 devices. The use of separate internal and external address prefixes creates complexity for DNS deployment, due the desire for internal nodes to communicate with other internal nodes using internal addresses, while external nodes need to obtain external addresses to communicate with the same nodes.
Typically, this results in the deployment of "split DNS", which has it's own set of architectural implications [Ref Needed].

---

3. Motivations

TOC

In defining the NAT66 mechanism, it is not our goal to encourage the implementation or use of NAT66. There are significant technical problems associated with the deployment of any type of NAT, and the IETF does not recommend the use of NAT. We strongly encourage anyone who is considering the implementation or deployment of NAT66 to read RFC 4864 [RFC4864] (Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, “Local Network Protection for IPv6,” May 2007.), and to carefully consider the alternatives described in that document, many of which will cause fewer problems than NAT66.
We are documenting NAT66 because we believe that some people will choose to implement and deploy IPv6 NAT, in spite of our recommendation not to do so. Some enterprises may choose to deploy IPv6 NAT to gain provider independent internal addressing or to simplify site

multihoming. Others may consider the trade-offs and choose IPv6 NAT as a topology hiding mechanism. In other cases, administrators may choose to deploy IPv6 NAT to parallel their IPv4 NAT-based network architecture. Our goal is to define an IPv6-to-IPv6 NAT mechanism, NAT66, that will minimize the negative impacts of IPv6 NAT, in the event that some implementers do choose to implement an IPv6 NAT mechanism, and some network administrators do choose to deploy it.

---

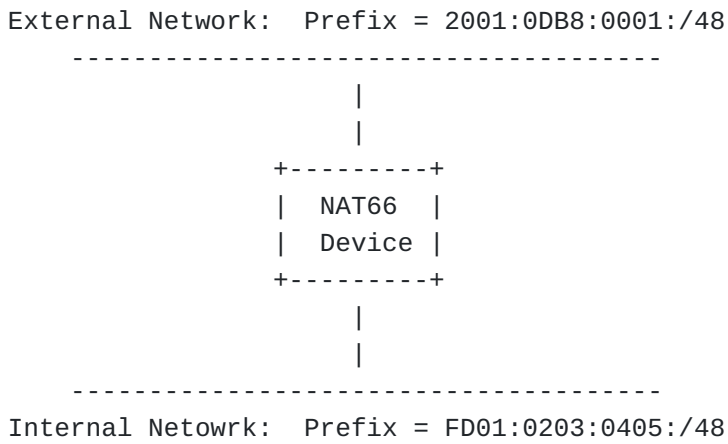
4. NAT66 Overview

TOC

NAT66 may be implemented in an IPv6 router to map one IPv6 address prefix to another IPv6 address prefix as each IPv6 packet transits the router. A router that implements a NAT66 function is referred to as a NAT66 device.

In its simplest form, a NAT66 device will be attached to two network links, one of which is an "internal" network link attached to a leaf network within a single administrative domain, and the other of which is an "external" network with connectivity to the global Internet. All of the hosts on the internal network will use addresses from a single, locally-routed prefix, and those addresses will be translated to/from addresses in a globally-routable prefix as IP packets transit the NAT66 device.

The following picture shows a NAT66 device attached to two networks. In this example, the internal network uses IPv6 Unique Local Addresses (ULAs) [RFC4193] (Hinden, R. and B. Haberman, “Unique Local IPv6 Unicast Addresses,” October 2005.) to represent the internal IPv6 nodes, and the external network uses globally routable IPv6 addresses to represent the same nodes.



When a NAT66 device forwards packets in the "outbound" direction, from the internal network to the external network, NAT66 overwrites the IPv6 source address (in the IPv6 header) with a corresponding address from the external prefix. When packets are forwarded in the "inbound" direction, from the external network to the internal network, the IPv6 destination address is overwritten with a corresponding address in the internal prefix. Using the prefixes shown in the diagram above, as an IP packet passes through the NAT66 device in the outbound direction, the source address prefix (FD01:0203:0405:/48) will be overwritten with the external address prefix (2001:0DB8:0001:/48). In an inbound packet, the destination prefix (2001:0DB8:0001:/48) will be overwritten with the internal network prefix (FD01:0203:0405:/48). In both cases, it is the local IPv6 address that is overwritten; the remote IPv6 address remains unchanged. Nodes on the internal network are said to be "behind" the NAT66 device.

NAT66 can also be used between two private networks. In these cases, both networks may use ULA prefixes, with each subnet in one network mapped into a corresponding subnet in the other network, and vice versa. Or, each network may use ULA prefixes for internal addressing and global unicast addresses on the other network. [TBD, add pictures of these examples].

In some cases, more than one NAT66 device may be attached to a network. In this case, they two NAT66 devices may be configured with the same internal and external prefixes, or they may be configured with the same internal prefix and different external prefixes. [TBD, add pictures of these examples.]

---

5. NAT66 Address Mapping Mechanisms

TOC

This document defines two address mapping functions that can be used in NAT66 devices. To comply with this specification, NAT66 devices MUST implement the Two-Way Algorithmic Address Mapping. NAT66 devices SHOULD implement the Topology Hiding Option.

The Two-Way Algorithmic Address Mapping mechanism and the Topology Hiding Option both use 1:1 (one-to-one) mappings, meaning that a given internal address is always mapped to the same external address.

When the Two-Way Algorithmic mapping is used, no per-node or per-flow state is maintained in the NAT66 box. Both inbound and outbound packets are translated algorithmically, using only information found in the IPv6 header. Due to this property, the Two-Way Algorithmic Address Mapping can support both outbound and inbound connection establishment without the need for state-priming or rendevous mechanisms. This is a significant improvement over NAT44 devices, but it also has significant security implications which are described in the Security Considerations section.

The Topology Hiding Option is intended to obscure the subnet information found in the internal IPv6 address prefix from external view, so that an external node cannot determine the structure of the internal network by looking at traffic outside of the NAT66 device. This features is called "topology hiding", and it is one of the benefits associated with NAT44. Because it is not possible to derive the full internal address simply by looking at the external address, the NAT66 device needs to maintain state in order to copy the correct internal address into inbound packets. This also means that inbound connection establishment will not work properly unless special provisions are made to enable inbound connectivity, such as configuring static state in the NAT66 device.

5.1. Checksum-Neutral Mapping

TOC

The NAT66 address mapping mechanisms described in this document are checksum-neutral, which means that they result in IP headers that will generate the same pseudo-header checksum when the checksum is calculated using the standard Internet checksum [RFC1071] (Braden, R., Borman, D., Partridge, C., and W. Plummer, “Computing the Internet checksum,” September 1988.). Any changes that are made during translation of the IPv6 prefix are offset by changes to other parts of the IPv6 address. This results in the transport layers (such as TCP and UDP) calculating the same IPv6 pseudo header checksum for both the internal and external forms of the same packet, which avoids the need for the NAT66 device to modify the transport layer headers.

The NAT66 address mapping mechanisms both use the same technique to ensure that they produce checksum-neutral transformations. When a changes is made to one of the fields in the IPv6 pseudo-header checksum, the checksum field in the transport layer header may become invalid. Fortunately, an incremental change in the area covered by the Internet standard checksum [RFC1071] (Braden, R., Borman, D., Partridge, C., and W. Plummer, “Computing the Internet checksum,” September 1988.) will result in a well-defined change to the checksum value [RFC1624] (Rijsinghani, A., “Computation of the Internet Checksum via Incremental Update,” May 1994.). So, a checksum change caused by modifying one part of the area covered by the checksum can be eliminated by making a complementary change to a different 16-bit field covered by the same checksum.

To produce a checksum neutral transformation, the NAT66 device calculates the 16-bit one's complement sum of the internal and external IPv6 prefixes. The difference between the original and mapped prefix checksums is calculated using 16-bit one's complement arithmetic, and the difference is added to a 16-bit value in another area of the local IPv6 address, thus resulting in an IPv6 header that will have the same pseudo-header checksum as the original header. Although the same

mechanism is used to ensure that both of the NAT66 mappings are checksum-neutral, there are differences in which parts of the IPv6 header are mapped and where the complementary change is made.

5.1.1. Two-Way Algorithmic Address Mapping

TOC

The Two-Way Algorithmic Address Mapping MUST be implemented on all NAT66 devices. This mapping consists of mapping an internal IPv6 prefix, typically a ULA, to/from an external prefix, typically a globally-routable unicast address, and making a complementary modification to 16 subnet bits in bits 49 through 64 of the local IPv6 addresss. The same transformation is performed in both the inbound and outbound directions, so the only state that is needed on the NAT66 box to peform this transformation is knowledge of the internal and external address prefixes in use.

For the network shown in the example diagram in the NAT66 Overview section above, we might have the following example:

Internal Prefix: FD01:0203:0405:/48 External Prefix: 2001:0DB8:0001:/48

If a node with internal address FD01:0203:0405:0001::1234 sends an outbound packet through the NAT66 device, the resulting external address will be 2001:0DB8:0001:D550::1234. The resulting address is obtained by calculating the checksum of both the internal and external 48-bit prefixes, sutracting the internal prefix from the external prefix using one's complement arithmetic and adding the result to the 16-bit subnet field (in this case 0x0001).

To show the work:

The one's complement checksum of FD01:0203:0405 is 0xFCF5. The one's complement checksum of 2001:0DB8:0001 is 0xD245. Using one's complement math, 0xD245 - 0xFCF5 = 0xD54F. The subnet mask in the original packet is 0x0001. Using one's complement math, 0x0001 + 0xD54F = 0xD550.

So, the value 0xD550 is written in the 16-bit subnet mask area, resulting in a mapped external address of 2001:0DB8:0001:D550::1234.

When a response packet is received, it will contain the destination address 2001:0DB8:0001:D550::0001, which will be mapped using the same mapping algorithm, back to FD01:0203:0405:0001::1234.

In this case, the difference between the two prefixes will be calculated as follows:

Using one's complement math, 0xFCF5 - 0xD245 = 0x2AB0. The subnet mask in the original packet = 0xD550. Using one's complement math, 0xD550 + 0x2AB0 = 0x0001.

So the value 0x0001 is written into the subnet field, and the internal value of the subnet field is properly restored.

This mapping results in no modification of the Interface Identifier (IID), which is held in the lower half of the IPv6 address, so it will not interfere with future protocols that may use unique IIDs for node identification.

Use of this mapping is restricted to cases where both the internal and external prefixes are 48 bits long (a /48) or shorter, leaving at least 16 subnet bits that can be modified to ensure checksum neutrality. This may not be a significant limitation in practice, because it is expected that most NAT66 devices will be used to map between a provider-allocated external prefix of /48 or shorter and a ULA that uses the same prefix length as the external prefix. In cases where one or both prefixes are longer than a /48, the Topology Hiding Option can be used.

5.1.2. Topology Hiding Option

TOC

The topology hiding option SHOULD be implemented on all NAT66 devices. It is very similar to the Two-Way Algorithmic Address mapping, except that the subnet bits in the destination address are mapped to zero in the outbound direction and are restored to their original value in the inbound direction. To remove the restriction on prefixes that have at least 16 bits of subnet space available, the checksum adjustment is made in the last 16 bits of the IP header, thus modifying the IPv6 Interface Identifier. Because the Interface Identifier may no longer be unique, the "u" bit [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) is cleared in the IID. This changes is also taken into account in the checksum adjustment.
For the network shown in the example diagram in the NAT66 Overview section above, we might have the following example:
Internal Prefix: FD01:0203:0405:/48 External Prefix: 2001:0DB8:0001:/48
If a node with internal address FD01:0203:0405:0001::1234 sends an outbound packet through the NAT66 device, the resulting external address will be 2001:0DB8:0001:0000:0002::E782. The resulting address is obtained by calculating the checksum of both the internal and external 48-bit prefixes and subtracting the internal prefix checsum from the external prefix checksum. If the "u" bit is cleared in the original address (the IID has universal scope), set the bit in the mapped address and add 0xFFFD to the checksum difference calculated above. Then, add the checksum difference to the value o the last 16 bits of the IPv6 address.
To show the work:
The one's complement checksum of FD01:0203:0405:0001 is 0xFCF4. The one's complement checksum of 2001:0DB8:0001:0000 is 0xD245. Using one's complement math, 0xD245 - 0xFCF4 = 0xD550. The original address has the "u" bit clear, so 0xD550 + 0xFFFD = 0xD54E. The last 16 bits of the original address are 0x1234. Using one's complement math, 0x1234 + 0xD54E = 0xE782.
So, when the prefix is mapped, the "u" bit is set in the IID, and the value 0xE782 is written into the last 16 bits of the address, this results in a mapped external address of 2001:0DB8:0001:0000:0002::E782.

When a response packet is received, it will contain the destination address 2001:0DB8:0001:0000:0002::E782. Unfortunately that address does not contain enough information to do an algorithmic reverse transformation, as the subnet bits were zeroed out when the external address was selected. Therefore, the NAT66 will need to consult its internal state to perform the reverse address mapping.

The internal state used for this mapping could consist of dynamic per-node mapping state, as is maintained in most NAT44 devices today, or it could consist of a static mapping of external addresses to internal addresses. If dynamic state is used, inbound connections to nodes that have not yet communicated externally will fail, because there will be no state to perform the inbound mapping. NAT66 implementations SHOULD provide a means for network administrators to configure static mapping state to allow inbound mapping when the Topology Hiding Option is in use.

Note: We could place the checksum adjustment in the 16-bit subnet field, if the prefixes are /48 or less, thus avoiding the need to modify the IID in those cases. Is that worth doing? We can't blindly overwrite the 16-bit following prefix no matter where they are, because of the need to maintain the "u" and "g" bits in the 7th and 8th bits of the IID.

---

6. Prefixes for Internal Addressing

TOC

IPv6 includes a form of local addressing called Unique Local Addresses (ULAs) [RFC4193] (Hinden, R. and B. Haberman, “Unique Local IPv6 Unicast Addresses,” October 2005.), and it is RECOMMENDED that ULAs be used to address network nodes that are located on an internal network serviced by a NAT66 device.

NAT66 devices MUST support manual configuration of internal and external address prefixes, and MUST NOT place any restrictions on those prefixes except that they be valid IPv6 unicast address prefixes, as described in [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.), and that they meet the requirements outlined in this document.

NAT66 devices that do not have a manually configured internal prefix SHOULD randomly generate a ULA prefix for the internal network and advertise that prefix in router advertisements. NAT66 boxes with more than one internal interface SHOULD assign a subnet number to each link, and include the subnet number in router advertisements on the corresponding link. NAT66 devices that generate a ULA prefix MUST generate the prefix using a random number as described in RFC4291 [RFC4193] (Hinden, R. and B. Haberman, “Unique Local IPv6 Unicast Addresses,” October 2005.), and SHOULD store the randomly generated prefix is non-volatile storage for continued use.

7. A Note on Port Mapping

TOC

In addition to overwriting IP addresses when packets are forwarded, NAT44 devices often overwrite the source port number in outbound traffic, and the destination port number in inbound traffic. This mechanism is called "port mapping".

The major benefit of port mapping, and perhaps its only significant benefit, is that it allows multiple computers to share a single IPv4 address. A large number of internal IPv4 addresses (typically from the 10.0.0.0/8 prefix) can be mapped into a single external, globally routable IPv4 address, with the local port number used to identify which internal node should receive each inbound packet. This address amplification feature should not be needed in IPv6, where every attached network should be assigned at least a /48 prefix, leaving room for 16 subnet bits and a 64 bit Interface Identifier [RFC3587] (Hinden, R., Deering, S., and E. Nordmark, “IPv6 Global Unicast Address Format,” August 2003.).

Since port mapping requires re-writing a portion of the transport layer header, it requires NAT66 devices to be aware of all of the transport protocols that they forward, thus stifling the development of new and improved transport protocols. Modifying the transport layer header is incompatible with security mechanisms that encrypt the full IP payload, and restricts the NAT66 device to forwarding trasnport layers that use weak checksum algorithms that are easily recalculated in routers. Since there is significant detriment caused by modifying transport layer headers and very little, if any, benefit to the use of port mapping in IPv6, NAT66 devices that comply with this specification MUST NOT perform port mapping.

8. Security Considerations

TOC

When NAT66 is deployed using the two-way, algorithmic address mapping, it allows direct inbound connections to internal nodes. While this can be viewed as a benefit of NAT66 vs. NAT44, it does open internal nodes to attacks that would not be possible in a NAT44 network. Although this situation is no worse, from a security standpoint, than running IPv6 with no NAT, some enterprises may assume that a NAT66 device will offer similar protection to a NAT44 device. For this reason, it is RECOMMENDED that NAT66 devices include an IPv6 firewall function, and the firewall function SHOULD be configured by default to block all incoming connections. Administrators could then enable inbound connectivity for specific ports by reconfiguring the firewall.

9. IANA Considerations

TOC

This document has no IANA considerations.

---

10. Acknowledgements

TOC

The checksum-neutral algorithmic address mapping described in this document is based on e-mail written by Iljtsch Van Beijnum. A similar mapping mechanism to the one described in this document was previously described in a document that can be found here: http://users.piuha.net/chvogt/pub/2008/vogt-2008-six-one-router-design.pdf. [TBD, move to an informative reference].
The following people provided advice or review comments that substantially improved this document: Ed Jankiewicz, .
This document was written using the xml2rfc tool described in RFC 2629 [RFC2629] (Rose, M., “Writing I-Ds and RFCs using XML,” June 1999.).

---

11. Change Log

TOC

11.1. Changes Between -00 and -01

TOC

There were several minor changes made between the -00 and -01 versions of this draft:

- *Added Fred Baker as a co-author.
- *Minor mathematical corrections.
- *Added AH to paragraph on NAT security issues.
- *Added additional NAT topologies to overview (diagrams TBD).

---

12. References

TOC

12.1. Normative References

TOC

[RFC1071]	Braden, R., Borman, D., Partridge, C., and W. Plummer, "Computing the Internet checksum," RFC 1071, September 1988 (TXT).
[RFC1624]	Rijsinghani, A., "Computation of the Internet Checksum via Incremental Update," RFC 1624, May 1994 (TXT).
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC3587]	Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format," RFC 3587, August 2003 (TXT).
[RFC4193]	Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses," RFC 4193, October 2005 (TXT).
[RFC4291]	Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291, February 2006 (TXT).

12.2. Informative References

TOC

[RFC2629]	Rose, M., "Writing I-Ds and RFCs using XML," RFC 2629, June 1999 (TXT, HTML, XML).
[RFC2993]	Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000 (TXT).
[RFC4864]	Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6," RFC 4864, May 2007 (TXT).

Authors' Addresses

TOC

	Margaret Wasserman
	Sandstorm Enterprises
	14 Summer Street
	Malden, MA 02148
	USA
Phone:	+1 781 333 3200
Email:	mrw@lilacglade.org
URI:	http://www.sandstorm.net
	Fred Baker
	Cisco Systems
	Santa Barbara, California 93117
	USA
Phone:	+1-408-526-4257

Email: fred@cisco.com

Full Copyright Statement

TOC

Intellectual Property