BEHAVE
Internet-Draft
Expires: April 15, 2005

F. Audet
Nortel Networks
C. Jennings
Cisco Systems

October 15, 2004

NAT Behavioral Requirements for Unicast UDP
draft-ietf-behave-nat-00

Status of this Memo

Copyright Notice

Abstract

This document defines basic terminology for describing different types of behavior for NATs when handling unicast UDP. It also defines a set of requirements for NATs that would allow many applications, such as multimedia communications or online gaming, to work consistently. Developing NATs that meet this set of requirements will greatly increase the likelihood that applications will function properly.

Table of Contents

1. Introduction

Network Address Translators (NAT) are well known to cause very significant problems with applications that carry IP addresses in the payload RFC 3027 [5]. Applications that suffer from this problem include Voice Over IP and Multimedia Over IP (e.g., SIP [6] and H.323 [11]), as well as online gaming.

Many techniques are used to attempt to make realtime multimedia applications, online games, and other applications work across NATs. Application Level Gateways [3] are one such mechanism. STUN [7] describes a UNilateral Self-Address Translation (UNSAF) mechanism [2] . UDP Relays have also been used to enable applications across NATs, but these are generally seen as a solution of last resort. ICE [9] describes a methodology for using many of these techniques and avoiding a UDP Relay unless the type of NAT is such that it forces the use of such a UDP Relay. This specification defines requirements for improving NATs. Meeting these requirements ensures that applications will not be forced to use UDP media relay.

Several recommendations regarding NATs for Peer-to-Peer media were made in [10] and this specification derives some of its requirements from that draft.

As pointed out in UNSAF [2] , "From observations of deployed networks, it is clear that different NAT boxes' implementation vary widely in terms of how they handle different traffic and addressing cases." This wide degree of variability is one part of what contributes to the overall brittleness introduced by NATs and makes it extremely difficult to predict how any given protocol will behave on a network traversing NATs. Discussions with many of the major NAT vendors have made it clear that they would prefer to deploy NATs that were deterministic and caused the least harm to applications while still meeting the requirements that caused their customers to deploy NATs in the first place. The problem the NAT vendors face is they are not sure how best to do that or how to document how their NATs behave.

The goals of this document are to define a set of common terminology for describing the behavior of NATs and to produce a set of requirements on a specific set of behaviors for NATs. The requirements represent what many vendors are already doing, and it is not expected that it should be any more difficult to build a NAT that meets these requirements or that these requirements should affect performance.

The authors strongly believe that if there were a common set of requirements that were simple and useful for voice, video, and games,

the bulk of the NAT vendors would choose to meet those requirements. This document will simplify the analysis of protocols for deciding whether or not they work in this environment and will allow providers of services that have NAT traversal issues to make statements about where their applications will work and where they will not, as well as to specify requirements for NATs.

1.1 Scope

This specification only covers Traditional NATs [4]. Bi-directional, Twice NAT, and Multihomed NAT [3] are outside the scope of this document.

Approaches using directly signaled control off the middle boxes such as Midcom, UPnP, or in-path signaling are out of scope.

UDP Relays are out of the scope of this document.

Application aspects are out of scope as the focus is strictly on the NAT itself.

This document only covers the UDP unicast aspects of NAT traversal and does not cover TCP, ICMP, IPSEC, or other protocols. Since the document is on UDP unicast only, packet inspection below the UDP layer (including RTP) is also out-of-scope.

This document does not cover Firewalls. However, it does cover the inherent filtering aspects of NATs.

2. Terminology

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 [1].

It is assumed that the reader is familiar with the terminology described in RFC 2663 [3] and RFC 3022 [4]. This specification attempts to preserve the terminology used in those RFCs.

This document uses the term "session" as defined in RFC 2663 [3]: "TCP/UDP sessions are uniquely identified by the tuple of (source IP address, source TCP/UDP ports, target IP address, target TCP/UDP Port)."

This document uses the term "address binding" as defined in RFC 2663 RFC 3022: "Address binding is the phase in which a local IP address is associated with an external address, or vice versa, for purpose of translation."

The term NAT is used to refer to both traditional address translation and address port translation. The authors understand that there was a time when these were considered different, but terminology has changed over time, and the term NAT has subsumed port translation as part of it.

RFC 3489 [7] defines a terminology for different NAT variations. In particular, it uses the terms "Full Cone", "Restricted Cone", "Port Restricted Cone" and "Symmetric" to refer to different variations of NATs. Unfortunately, this terminology has been the source of much confusion. This terminology does not distinguish between the NAT behavior and the filtering behavior of the device. It was found that many devices' behaviors do not exactly fit into the described variations. For example, a device could be symmetric from a filtering point of view and Cone from a NAT point of view. Other aspects of NATs are not covered by this terminology: for example, many NATs will switch over from basic NAT (preserving ports) to NAPT (mapping ports) in order to preserve ports when possible.

This specification will therefore not use the Cone/Symmetric terminology. Furthermore, many other important behaviors are not fully described by the Cone/Symmetric terminology. This specification refers to specific individual NAT behaviors instead of using the Cone/Symmetric terminology.

Note: RFC 3489 defines a "Symmetric NAT" in effectively two parts:

1. All requests from the same internal IP address and port to a specific destination IP address and port are mapped to the same external IP address and port. If a host sends a packet with the same source address and port to different destination addresses or ports, a different mapping is used for each.
2. Furthermore, only the external host that receives a packet can send a UDP packet back to the internal host.

Condition 1 is the NAT behavior and condition 2 is the filtering behavior. However, they are not necessarily dependent: we have observed NATs that will conform to condition (1) but not to (2). Using RFC 3489, this type of NAT would be detected as a "Cone NAT" since it uses condition (2). Using a different algorithm such as the one described in NATCHECK [12] which uses condition (1), the same NAT would be detected as a "Symmetric NAT". If the endpoint receiving the media has a permissive policy on accepting media, condition (2) is more appropriate, but if it has a restrictive policy, condition (1) is more appropriate.

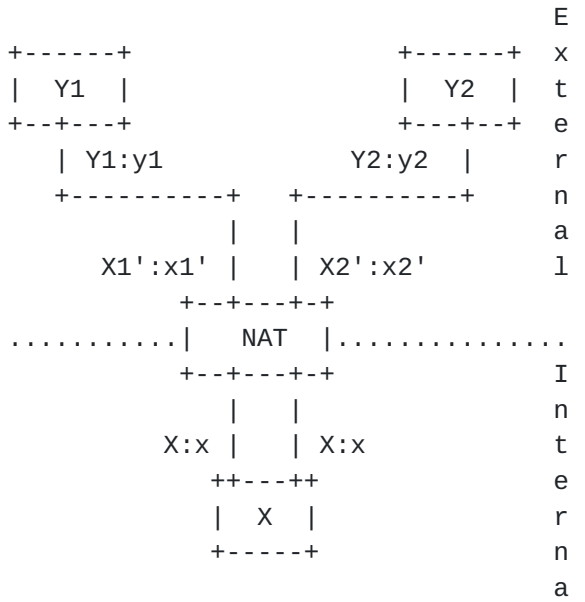
3. Network Address and Port Translation Behavior

This section describes the various NAT behaviors applicable to dynamic NAT; static NAT is outside the scope of this document.

3.1 Address and Port Binding

When an internal endpoint opens an outgoing UDP session through a NAT, the NAT assigns the session an external IP address and port number so that subsequent response packets from the external endpoint can be received by the NAT, translated and forwarded to the internal endpoint. This is a binding between an internal IP address and port (IP:port) and external IP:port tuple. It establishes the translation that will be performed by the NAT for the duration of the session. For many applications, it is important to distinguish the behavior of the NAT when there are multiple simultaneous sessions established to different external endpoints.

The key behavior to describe is the criteria for re-use of a binding for new sessions to external endpoints, after establishing a first binding between an internal X:x address and port and an external Y1:y1 address tuple. Let's assume that the internal IP address and port X:x is mapped to X1':x1' for this first session. The endpoint then sends from X:x to an external address Y2:y2 and gets a mapping of X2':x2' on the NAT. The relationship between X1':x1' and X2':x2' for various combinations of the relationship between Y1:y1 and Y2:y2 is critical for describing the NAT behavior. This arrangement is illustrated in the following diagram:



The following address and port binding behavior are defined:

External NAT binding is endpoint independent: The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port (X:x) to any external IP address and port. Specifically, X1':x1' equals X2':x2' for all values of Y2:y2. (From an RFC 3489 NAT perspective, this is a "Cone NAT" where the sub-type is really based on the filtering behavior.)

External NAT binding is endpoint address dependent: The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port (X:x) only for sessions to the same external IP address, regardless of the external port. Specifically, X1':x1' equals X2':x2' if, and only if, Y2 equals Y1. (From an RFC 3489 NAT perspective, but not necessarily a filtering perspective, this is a "Symmetric NAT".)

External NAT binding is endpoint address and port dependent: The NAT reuses the port binding for subsequent sessions initiated from the same internal IP address and port (X:x) only for sessions to the same external and port. Specifically, X1':x1' equals X2':x2' if, and only if, Y2:y2 equals Y1:y1. (From an RFC 3489 NAT perspective, but not necessarily a filtering perspective, this is a "Symmetric NAT".)

The three possibilities are abbreviated as NB=I, NB=AD, and NB=APD, respectively. NB stands for Nat Binding, I for independent, AD for Address Dependent, and APD for Address Port Dependent.

It is important to note that these three possible choices make no difference to the security properties of the NAT. The security properties are fully determined by which packets the NAT allows in and which it does not. This is determined by the filtering behavior in the filtering portions of the NAT.

Some NATs are capable of assigning IP addresses from a pool of IP addresses on the external side of the NAT, as opposed to just a single IP address. This is especially common with larger NATs. Some NATs use the external IP address binding in an arbitrary fashion (i.e. randomly): one internal IP address could have multiple external IP address bindings active at the same time for different sessions. These NATs have an "IP address pooling" behavior of "arbitrary". Other NATs use the same external IP address binding for all sessions associated with the same internal IP address in any circumstance, even if ports are running out, in which case they would

fail to establish a session. These NATs have an "IP address pooling" behavior of "Paired, strict." Yet other NATs use the same external IP address binding for all sessions associated with the same IP internal IP address only when possible. These NATs have an "IP address pooling" behavior of "Paired, best effort." NATs that use an "IP address pooling" behavior of "arbitrary" can cause issues for applications that use multiple ports from the same endpoint but have no way to negotiate IP addresses individually (e.g., RTP and RTCP ports).

3.2 Port Assignment

Some NATs attempt to preserve the port number used internally when assigning a binding to an external IP address and port (e.g., X:x to X':x). A basic NAT, for example, will preserve the same port and will assign a different IP address from a pool of external IP addresses in case of port collision (e.g. X1:x to X1':x and X2:x to X2':x). This is only possible as long as the NAT has enough external IP addresses. If the port x is already in use on all available external IP addresses, then the NAT needs to switch from Basic NAT to a Network Address and Port Translator (NAPT) mode (i.e., X1:x to X':x and X2:x to X':x'). This is referred to as "port preservation". It does not guarantee that the external port x' will always be the same as the internal port x but only that the NAT will preserve the port if possible.

A NAT that does not attempt to make the external port numbers match the internal port numbers in any case (i.e., X1:x to X':x1', X2:x to X':x2') is referred to as "no port preservation".

Tools such as network sniffers identify traffic based on the destination port, not the source port, so port preservation does not help these tools.

Some particularly nasty NATs use port overloading, i.e. they always use port preservation even in the case of collision (i.e., X1:x to X':x, and X2:x to X':x). These NATs rely on the source of the response from the external endpoint (Y:y, Z:z) to forward a packet to the proper internal endpoint (X1 or X2). Port overloading fails if the two internal endpoints are establishing sessions to the same external destination. This is referred to as "port overloaded".

Most applications fail in some cases with "Port overloaded". It is clear that "port overloaded" behavior will result in many problems.

Some NATs preserve the parity of the UDP port, i.e., an even port will be mapped to an even port, and an odd port will be mapped to an odd port. This behavior respects the RFC 3550 [8] rule that RTP use

even ports, and RTCP use odd ports. There is some very unfortunate wording in RFC 3550 section 11, which can cause some problematic behavior in RTP clients. The wording could be interpreted as saying that if a client receives an odd port for sending RTP, it SHOULD change it to the next lower even number. This would obviously result in the loss of RTP/UDP. In the event that a NAT supporting port parity preservation is running out of available ports for a specific parity, it may either fail to assign a port, or it could decide not respect port parity. We refer to these properties as "port parity preservation" of "No", "Yes, strict" and "Yes, best effort".

When NATs do allocate a new source port, there is the issue of which IANA-defined range of port to choose. The ranges are "well-known" from 0 to 1023, "registered" from 1024 to 49151, and "dynamic/ private" from 49152 through 65535. For most protocols, these are destination ports and not source ports, so mapping a source port to a source port that is already registered is unlikely to have any bad effects. There has been some suggestion that some IDS systems with deep packet inspection devices may end up looking at the ports. This is clearly true for destination ports but less understood for source ports. Some NATs may choose to use only the ports in the dynamic range; the only down side of this practice is that it limits the number of ports available. Other NAT devices may use everything but the well-known range and may prefer to use the dynamics range first or possibly avoid the actual registered ports in the registered range.

3.3 Bind Refresh Direction

NAT UDP binding timeout implementations vary but include the timer's value and the way the binding timer is refreshed to keep the binding alive.

The binding timer is defined as the time a binding will stay active without packets traversing the NAT. There is great variation in the values used by different NATs.

Some NATs keep the binding active (i.e., refresh the timer value) when a packet goes from the internal side of the NAT to the external side of the NAT. This is referred to as having an Outbound NAT refresh behavior of "True".

Some NATs keep the binding active when a packet goes from the external side of the NAT to the internal side of the NAT. This is referred to as having an Inbound NAT refresh direction behavior of "True".

Some NATs keep the binding active on both, in which case both

properties are "True".

3.4 Bind Refresh Scope

If the binding is refreshed for all sessions on that bind by any outbound traffic, the NAT is said to have a NAT refresh method behavior of "Per binding". If the binding is refreshed only on a specific session on that particular bind by any outbound traffic, the NAT is said to have a "Per session" NAT refresh method behavior.

4. Filtering Behavior

This section describes various filtering behaviors observed in NATs.

4.1 Filtering of Unsolicited Packets

When an internal endpoint opens an outgoing UDP session through a NAT, the NAT assigns a filtering rule for the binding between an internal IP:port (X:x) and external IP:port (Y:y) tuple.

The key behavior to describe is what criteria are used by the NAT to filter packets originating from specific external endpoints.

External filtering is open: The NAT does not filter any packets.

External filtering is endpoint independent: The NAT filters out only packets not destined to the internal address and port X:x, regardless of the external IP address and port source (Z:z). The NAT forwards any packets destined to X:x. In other words, sending packets from the internal side of the NAT to any external IP address is sufficient to allow any packets back to the internal endpoint. (From an RFC 3489 filtering perspective, this is a "Full Cone NAT".)

External filtering is endpoint address dependent: The NAT filters out packets not destined to the internal address X:x. Additionally, the NAT will filter out packets from Y:y destined for the internal endpoint X:x if X:x has not sent packets to Y previously (independently of the port used by Y). In other words, for receiving packets from a specific external endpoint, it is necessary for the internal endpoint to send packets first to that specific external endpoint's IP address. (From an RFC 3489 filtering perspective, this is a "Restricted Cone NAT".)

External filtering is endpoint address and port dependent: This is similar to the previous behavior, except that the external port is also relevant. The NAT filters out packets not destined for the internal address X:x. Additionally, the NAT will filter out

packets from Y:y destined for the internal endpoint X:x if X:x has not sent packets to Y:y previously. In other words, for receiving packets from a specific external endpoint, it is necessary for the internal endpoint to send packets first to that external endpoint's IP address and port. (From an RFC 3489 filtering perspective, this is both a "Port Restricted Cone NAT" and a "Symmetric NAT" as they have the same filtering behavior.)

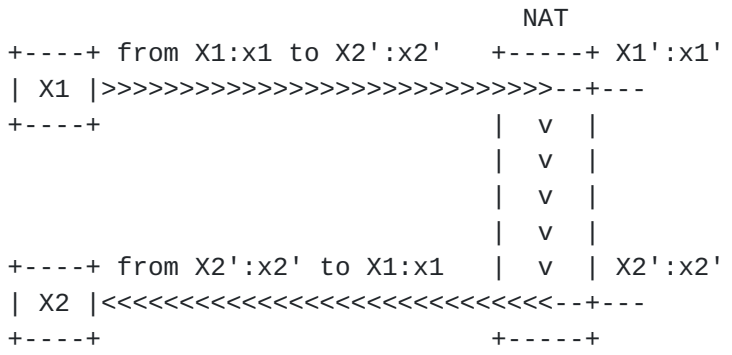
These are abbreviated as EF=O, EF=I, EF=AD, EF=APD respectively. In the case of a NAT, "open" cannot forward a packet unless there is a NAT binding, so for all practical purposes, a NAT will never be "open" but will be one of the others.

4.2 NAT Filter Refresh

The time for which a NAT filter is valid can be refreshed based on packets that are inbound, outbound, or going either direction. In the case of EF=AD or EF=APD NATs, the scope of the refresh could include the filters for just the particular port and destination or for all the ports and destinations sharing the same external address and port on the NAT.

5. Hairpinning Behavior

If two hosts (called X1 and X2) are behind the same NAT and exchanging traffic, the NAT may allocate an address on the outside of the NAT for X2, called X2':x2'. If X1 sends traffic to X2':x2', it goes to the NAT, which must relay the traffic from X1 to X2. This is referred to as hairpinning and is illustrated below.



Hairpinning allows two endpoints on the internal side of the NAT to communicate even if they only use each other's external IP addresses and ports.

More formally, a NAT that supports hairpinning forwards packets

originating from an internal address, X1:x1, destined for an external address X2':x2' that has an active binding to an internal address X2:x2, back to that internal address X2:x2. (Note that typically, X1'=X2'.)

Furthermore, the NAT may present the hairpinned packet with either an internal or an external source IP address and port. The hairpinning NAT behavior can therefore be either "External source IP address and port" or "Internal source IP address and port". "Internal source IP address and port" may cause problems by confusing an implementation that is expecting an external IP address and port.

6. Application Level Gateways

Certain NATs have implemented Application Level Gateways (ALGs) for various protocols, including protocols for negotiating peer-to-peer UDP sessions.

Certain NATs have these ALGs turned on permanently, others have them turned on by default but let them be be turned off, and others have them turned off by default but let them be turned on.

NAT ALGs may interfere with UNSAF methods and must therefore be used with extreme caution.

7. Deterministic Properties

The diagnosis is further complicated by the fact that under some conditions the same NAT will exhibit different behaviors. This has been seen on NATs that preserve ports or have specific algorithms for selecting a port other than a free one. If the external port that the NAT wishes to use is already in use by another session, the NAT must select a different port. This results in different code paths for this conflict case, which results in different behavior.

For example, if three hosts X1, X2, and X3 all send from the same port x, through a port preserving NAT with only one external IP address, called X1', the first one to send (i.e., X1) will get an external port of x but the next two will get x2' and x3' (where these are not equal to x). There are NATs where the External NAT Binding characteristics and the External Filter characteristics change between the X1:x and the X2:x binding. To make matters worse, there are NATs where the behavior may be the same on the X1:x and X2:x binds but different on the third X3:x binding.

Some NATs that try to reuse external ports flow from two internal IP addresses to two different external IP addresses. For example, X1:x is going to Y1:y1 and X2:x is going to Y2:y2, where Y1:y1 does not

equal Y2:y2. Some NATs will map X1:x to X1':x and will also map X2:x to X1':x. This works in the case where the NAT Binding is address port dependent. However some NATs change their behavior when this type of port reuse is happening. The NAT may look like it has NAT Bindings that are independent when this type of reuse is not happening but may change to Address Port Dependent when this reuse happens.

Any NAT that changes the NAT Binding or the External Filtering at any point in time or under any particular conditions is referred to as a "non-deterministic" NAT. NATs that don't are called "deterministic".

Non-deterministic NATs generally change behavior when a conflict of some sort happens, i.e. when the port that would normally be used is already in use by another bind. The NAT binding and External Filtering in the absence of conflict is referred to as the Primary behavior. The behavior after the first conflict is referred to as Secondary and after the second conflict is referred to as Tertiary. No NATs have been observed that change on further conflicts but additional testing may be required.

8. ICMP Behavior

There are cases in which a host inside the NAT sends a packet to the NAT that gets relayed towards a host on the external side of the NAT that results in an ICMP Destination Unreachable message being returned to the NAT. Most NATs will send an appropriate ICMP Destination Unreachable message to the internal host that sent the original packet. NATs that do not filter out this ICMP Destination Unreachable message when it is in reply to a IP packet sent are referred to as "Support Destination Unreachable" (abbreviated SU).

Incoming Destination Unreachable messages can be ignored after some period of time after the packet which elicited the Destination Unreachable message. This IMCP timeout needs to be greater than the RTT for any destination the NAT may attempt to send IP packets to. Keep in mind satellite links when setting this timeout.

Applications use the destination unreachable message to decide that they can stop trying to retransmit to a particular IP address and can fail over to a secondary address. If a destination unreachable message is not received, the fail over will take too long for many applications. Another key use of this message is for MTU discovery (described in RFC 1191 [14]). MTU discovery is important for allowing applications to avoid the fragmentation problems discussed in the next section. The arrival of a destination unreachable packet cannot destroy the NAT binding, as the occasional arrival of these messages is normal for black-hole discovery RFC 2923 [17].

There is no significant security advantage to blocking these ICMP Destination Unreachable packets.

9. Fragmentation of Packets

When sending a packet, there are two situations that may cause IP fragmentation for packets from the inside to the outside. It is worth noting that many IP stacks do not use Path MTU Discovery with UDP packets.

9.1 Smaller Adjacent MTU

The first situation is when the MTU of the adjacent link is too small. This can occur if the NAT is doing PPPoE, or if the NAT has been configured with a small MTU to reduce serialization delay when sending large packets and small, higher-priority packets.

The packet could have its Don't Fragment bit set to 1 (DF=1) or 0 (DF=0).

If the packet has DF=1, the NAT should send back an ICMP message "fragmentation needed and DF set" message to the host RFC 792 [18].

If the packet has DF=0, the NAT should fragment the packet and send the fragments, in order. This is the same function a router performs in a similar situation RFC 1812 [19].

NATs that operate as described in this section are described as "Supports Fragmentation" (abbreviated SF).

9.2 Smaller Network MTU

The second situation is when the MTU in the middle of the network is too small. If DF=0, the router adjacent to the small-MTU segment will fragment the packet and forward the fragments RFC 1812 [19].

If DF=1, the router adjacent to the small-MTU segment will send the ICMP message "fragmentation needed and DF set" back towards the NAT. The NAT needs to forward this ICMP message to the inside host.

The classification of NATs that perform this behavior is covered in the ICMP section of this document.

10. Receiving Fragmented Packets

For a variety of reasons, a NAT may receive a fragmented UDP packet. The IP packet containing the UDP header could arrive first or last depending on network conditions, packet ordering, and the

implementation of the IP stack that generated the fragments.

A NAT that is capable only of receiving UDP fragments in order (that is, with the UDP header in the first packet) and forwarding each of the fragments to the internal host is described as Received Fragments Ordered (abbreviated RF=O).

A NAT that is capable of receiving UDP fragments in or out of order and forwarding the individual packets (or a reassembled packet) to the internal host is referred to as Receive Fragments Out of Order (abbreviated RF=OO). See the Security Considerations section of this document for a discussion of this behavior.

A NAT that is neither of these is referred to as Receive Fragments None (abbreviated RF=N).

11. Requirements

The requirements in this section are aimed at minimizing the damage caused by NATs to applications such as realtime communications and online gaming.

It should be understood, however, that applications normally do not know in advance if the NAT conforms to the recommendations defined in this section. Peer-to-peer media applications still need to use normal procedures such as ICE [9].

REQ-1: A NAT MUST have an "External NAT Binding is endpoint independent" behavior (NB=I).

REQ-2: It is RECOMMENDED that a NAT have an "IP address pooling" behavior of "Paired, best effort". Note that this requirement is not applicable to NATs that do not support IP address pooling.

REQ-2a: A NAT MAY have an "IP address pooling" behavior of "Yes, strict."
REQ-2b: A NAT MUST NOT have an "IP address pooling" behavior of "No".

REQ-3: It is RECOMMENDED that a NAT have a "No port preservation" behavior.

REQ-3a: A NAT MAY use a "Port preservation" behavior.
REQ-3b: A NAT MUST NOT have a "Port overloaded" behavior.
REQ-3c: If the NAT changes the source port, it MUST NOT allocate the new port from within the range of 0-1023.

REQ-4: It is RECOMMENDED that a NAT have a "Port parity preservation" behavior of "Yes, best effort".

REQ-4a: A NAT MAY have a "Port parity preservation" behavior of "Yes, strict."
REQ-4b: A NAT MUST NOT have a "Port parity preservation" behavior of "No".

REQ-5: A dynamic NAT UDP binding timer MUST NOT expire in less than 2 minutes.

REQ-5a: The value of the NAT UDP binding timer MAY be configurable.
REQ-5b: A default value of 5 minutes for the NAT UDP binding timer is RECOMMENDED.

REQ-6: The NAT UDP timeout binding MUST have a NAT Outbound refresh behavior of "True".

REQ-6a: The NAT UDP timeout binding MAY have a NAT Inbound refresh behavior of "True".
REQ-6b: The NAT UDP timeout binding MUST have a NAT refresh method behavior of "Per binding" (i.e. refresh all sessions active on a particular bind).

REQ-7: It is RECOMMENDED that a NAT have an "External filtering is endpoint address dependent" behavior. (EF=AD)

REQ-7a: A NAT MAY have an "External filtering is endpoint independent" behavior. (EF=I)
REQ-7b: A NAT MAY have an "External filtering is endpoint address and port dependent" behavior. (EF=APD)

REQ-8: The NAT UDP filter timeout behavior MUST be the same as the NAT UDP binding timeout.

REQ-9: A NAT MUST support "Hairpinning" behavior.

REQ-9a: A NAT Hairpinning behavior MUST be "External source IP address and port".

REQ-10: A NAT MUST have the capability to turn off individually all ALGs it supports, except for DNS and IPsec.

REQ-10a: Any NAT ALG for SIP MUST be turned off by default.

REQ-11: A NAT MUST have deterministic behavior.

REQ-12: A NAT MUST support ICMP Destination Unreachable (SU).

REQ-12a: The ICMP timeout SHOULD be greater than 2 seconds.
REQ-13: A NAT MUST support fragmentation of packets larger than link MTU (SF).

REQ-14: A NAT MUST support receiving in order fragments, so it MUST be RF=O or RF=OO.

REQ-14a: A NAT MAY support receiving fragmented packets that are out of order and be of type RF=OO.

11.1 Requirement Discussion

This section describes why each of these requirements was chosen and the consequences of violating any of them:

REQ-1: In order for UNSAF methods to work, REQ-1 needs to be met. Failure to meet REQ-1 will force the use of a Media Relay which is very often impractical.

REQ-2: This will allow applications that use multiple ports originating from the same internal IP address to also have the same external IP address, but without running out of ports unnecessarily.

REQ-3: NATs that implement port preservation have to deal with conflicts on ports, and the multiple code paths this introduces often result in nondeterministic behavior.

REQ-3a: Port preservation can work, but the NAT implementors need to be very careful that it does not become a nondeterministic NAT.
REQ-3b: REQ-2b must be met in order to enable two applications on the internal side of the NAT both to use the same port to try to communicate with the same destination.
REQ-3c: This requirement is because some applications may not be able to use ports in the well-known range because of priviledge restrictions.

REQ-4: This will respect the RTP/RTCP parity convention when possible, but without running out of ports unnecessarily.

REQ-5: This requirement is to ensure that the timeout is long enough to avoid too frequent timer refresh packets.

REQ-5a: Configuration is desirable for adapting to specific networks and troubleshooting.
REQ-5b: This default is to avoid too frequent timer refresh packets.

REQ-6: Outbound refresh is necessary for allowing the client to keep the binding alive.

REQ-6a: Inbound refresh may be useful for applications where there is no outgoing UDP traffic.
REQ-6b: Using the refresh on a per binding basis avoids the need for separate keep alives for all the available sessions.

REQ-7: Filtering based on the IP address is felt to have the maximum balance between security and usefulness. See below.

REQ-7a: Filtering independently of the external IP address and port is not as secure: an unauthorized packet could get at a specific port while the port was kept open if it was lucky enough to find the port open.
REQ-7b: In theory, filtering based on both IP address and port is more secure than filtering based only on the IP address (because the external endpoint could in reality be two endpoints behind another NAT, where one of the two endpoints is an attacker). However, such a restrictive policy could interfere with certain applications that use more than one port.

REQ-8: This is to avoid overly complex applications.

REQ-9: This requirement is to allow communications between two endpoints behind the same NAT when they are trying each other's external IP addresses.

REQ-9a: Using the external IP address is necessary for applications with a restrictive policy of not accepting packets from IP addresses that differ from what is expected.

REQ-10: NAT ALGs may interfere with UNSAF methods.

REQ-10a: A SIP NAT ALG will interfere with UNSAF methods.

REQ-11: Non-deterministic NATs are very difficult to troubleshoot because they require more intensive testing. This non-deterministic behavior is the root cause of much of the uncertainty that NATs introduce about whether or not applications will work.

REQ-12: This is easy to do, is used for many things including MTU discovery and rapid detection of error conditions, and has no negative consequences.

REQ-13 and REQ-13: Fragmented packets become more common with large video packets and should continue to work. Applications can use MTU discovery to work around this.

12. Security Considerations

NATs are often deployed to achieve security goals. Most of the recommendations and requirements in this document do not affect the security properties of these devices, but a few of them do have security implications and are discussed in this section.

This work recommends that the timers for binding be refreshed only on outgoing packets and does not make recommendations about whether or not inbound packets should update the timers. If inbound packets update the timers, an external attacker can keep the binding alive forever and attack future devices that may end up with the same internal address. A device that was also the DHCP server for the private address space could mitigate this by cleaning any bindings when a DHCP lease expired. For unicast UDP traffic (the scope of this document), it may not seem relevant to support inbound timer refresh; however, for multicast UDP, the question is harder. It is expected that future documents discussing NAT behavior with multicast traffic will refine the requirements around handling of the inbound refresh timer. Some devices today do update the timers on inbound packets.

This work recommends that the NAT filters be specific to the external IP only and not the external IP and port. It can be argued that this is less secure than using the IP and port. Devices that wish to filter on IP and port do still comply with these requirements.

Non-deterministic NATs are risky from a security point of view. They are very difficult to test because they are, well, non-deterministic. Testing by a person configuring one may result in the person thinking it is behaving as desired, yet under different conditions, which an attacker can create, the NAT may behave differently. These requirements recommend that devices be deterministic.

The work requires that NATs have an "external NAT binding is endpoint independent" behavior. This does not reduce the security of devices. Which packets are allowed to flow across the device is determined by the external filtering behavior, which is independent of the binding

behavior.

When a fragmented packet is received from the external side and the packets are out of order so that the initial fragment does not arrive first, many systems simply discard the out of order packets. Moreover, since some networks deliver small packets ahead of large ones, there can be many out of order fragments. NATs that are capable of delivering these out of order packets are possible but they need to store the out of order fragments, which can open up a DoS opportunity. Fragmentation has been a tool used in many attacks, some involving passing fragmented packets through NATs and others involving DoS attacks based on the state needed to reassemble the fragments. NAT implementers should be aware of RFC 3128 [16] and RFC 1858 [15].

13. IANA Considerations

There are no IANA considerations.

14. IAB Considerations

The IAB has studied the problem of "Unilateral Self Address Fixing", which is the general process by which a client attempts to determine its address in another realm on the other side of a NAT through a collaborative protocol reflection mechanism [2].

This specification does not in itself constitute an UNSAF application. It consists of a series of requirements for NATs aimed at minimizing the negative impact that those devices have on peer-to-peer media applications, especially when those applications are using UNSAF methods.

Section 3 of UNSAF lists several practical issues with solutions to NAT problems. This document makes recommendations to reduce the uncertainty and problems introduced by these practical issues with NATs. In addition, UNSAF lists five architectural considerations. Although this is not an UNSAF proposal, it is interesting to consider the impact of this work on these architectural considerations.

Arch-1: The scope of this is limited to UDP packets in NATs like the ones widely deployed today. The "fix" helps constrain the variability of NATs for true UNSAF solutions such as STUN.

Arch-2: This will exit at the same rate that NATs exit. It does not imply any protocol machinery that would continue to live after NATs were gone or make it more difficult to remove them.

Arch-3: This does not reduce the overall brittleness of NATs but will hopefully reduce some of the more outrageous NAT behaviors and make it easer to discuss and predict NAT behavior in given situations.

Arch-4: This work and the results [13] of various NATs represent the most comprehensive work at IETF on what the real issues are with NATs for applications like VoIP. This work and STUN have pointed out more than anything else the brittleness NATs introduce and the difficulty of addressing these issues.

Arch-5: This work and the test results [13] provide a reference model for what any UNSAF proposal might encounter in deployed NATs.

15. Acknowledgments

Dan Wing contributed substantial text on IP fragmentation.

The editor would like to acknowledge Bryan Ford, Pyda Srisuresh and Dan Kegel for the NATP2P [10] draft, from which a lot of the material in this specification is derived. Thanks to Rohan Mahy, Jonathan Rosenberg, Mary Barnes, Dan Wing, Melinda Shore, Lyndsay Campbell, Geoff Huston, Jiri Kuthan, Harald Welte and Spencer Dawkins for their important contributions.

16. References

16.1 Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

16.2 Informational References

[3] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[4] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[5] Holdrege, M. and P. Srisuresh, "Protocol Complications with the IP Network Address Translator", RFC 3027, January 2001.

[6] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[7] Rosenberg, J., Weinberger, J., Huitema, C. and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[8] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[9] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)", draft-ietf-mmusic-ice-02 (work in progress), July 2004.

[10] Ford, B., Srisuresh, P. and D. Kegel, "Peer-to-Peer(P2P) communication across Network Address Translators(NATs)", draft-ford-midcom-p2p-03 (work in progress), June 2004.

[11] "Packet-based Multimedia Communications Systems", ITU-T Recommendation H.323, July 2003.

[12] Ford, B. and D. Andersen, "Nat Check Web Site: http://midcom-p2p.sourceforge.net", June 2004.

[13] Jennings, C., "NAT Classification Results using STUN", draft-jennings-midcom-stun-results-01 (work in progress), July 2004.

[14] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[15] Ziemba, G., Reed, D. and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, October 1995.

[16] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, June 2001.

[17] Reynolds, J. and J. Postel, "Assigned numbers", RFC 923, October 1984.

[18] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.

[19] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

Authors' Addresses

Francois Audet
Nortel Networks
4655 Great America Parkway
Santa Clara, CA 95054
USA

Phone: +1-408-495-3756
EMail: audet@nortelnetworks.com

Cullen Jennings
Cisco Systems
170 West Tasman Drive
MS: SJC-21/2
San Jose, CA 95134
USA

Phone: +1-408-902-3341
EMail: fluffy@cisco.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.