NAT Behavioral Requirements for Unicast UDP
draft-ietf-behave-nat-udp-06

Status of this Memo

Copyright Notice

Abstract

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

Table of Contents

1. Applicability Statement

The purpose of this specification is to define a set of requirements
for NATs that would allow many applications, such as multimedia
communications or on-line gaming, to work consistently. Developing
NATs that meet this set of requirements will greatly increase the
likelihood that these applications will function properly.

The requirements of this specification apply to Traditional NATs as
described in RFC 2663 [8].

This document is meant to cover NATs of any size, from small
residential NATs to large Enterprise NATs. However, it should be
understood that Enterprise NATs normally provide much more than just
NAT capabilities: for example, they typically provide firewall
functionalities. Firewalls are specifically out-of-scope for this
specification; however, this specification does cover the inherent
filtering aspects of NATs.

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

UDP Relays are out-of-scope.

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

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

2. Introduction

Network Address Translators (NATs) are well known to cause very
significant problems with applications that carry IP addresses in the
payload RFC 3027 [10]. Applications that suffer from this problem
include Voice Over IP and Multimedia Over IP (e.g., SIP [12] and
H.323 [20]), 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 [8] are one such mechanism. STUN [17]
describes a UNilateral Self-Address Translation (UNSAF) mechanism
[15]. UDP Relays have also been used to enable applications across
NATs, but these are generally seen as a solution of last resort. ICE
[18] 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.

As pointed out in UNSAF [15], "From observations of deployed
networks, it is clear that different NAT box implementations vary
widely in terms of how they handle different traffic and addressing
cases." This wide degree of variability is one factor in the overall
brittleness introduced by NATs and makes it extremely difficult to
predict how any given protocol will behave on a network traversing
NAT. 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 NAT vendors face is that 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.

This document forms a common set of requirements that are simple and
useful for voice, video, and games, which can be implemented by NAT
vendors. 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 their own NAT requirements.

3. **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].

Readers are urged to refer to RFC 2263 [8] for information on NAT
taxonomy and terminology. Traditional NAT is the most common type of
NAT device deployed. Readers may refer to RFC 3022 [9] for detailed
information on traditional NAT. Traditional NAT has two main
varieties - Basic NAT and Network Address/Port Translator (NAPT).

NAPT is by far the most commonly deployed NAT device. NAPT allows

multiple internal hosts to share a single public IP address
simultaneously. When an internal host opens an outgoing TCP or UDP
session through a NAPT, the NAPT assigns the session a public IP
address and port number, so that subsequent response packets from the
external endpoint can be received by the NAPT, translated, and
forwarded to the internal host. The effect is that the NAPT
establishes a NAT session to translate the (private IP address,
private port number) tuple to (public IP address, public port number)
tuple and vice versa for the duration of the session. An issue of
relevance to peer-to-peer applications is how the NAT behaves when an
internal host initiates multiple simultaneous sessions from a single
(private IP, private port) endpoint to multiple distinct endpoints on
the external network. In this specification, the term "NAT" refers
to both "Basic NAT" and "Network Address/Port Translator (NAPT)".

This document uses the term "session" as defined in RFC 2663: "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 and port mapping" as the
translation between an external address and port and an internal
address and port. Note that this is not the same as an "address
binding" as defined in RFC 2663.

RFC 3489 [13] used the terms "Full Cone", "Restricted Cone", "Port
Restricted Cone" and "Symmetric" to refer to different variations of
NATs applicable to UDP only. Unfortunately, this terminology has
been the source of much confusion as it has proven inadequate at
describing real-life NAT behavior. This specification therefore
refers to specific individual NAT behaviors instead of using the
Cone/Symmetric terminology.

4. Network Address and Port Translation Behavior

This section describes the various NAT behaviors applicable to NATs.

4.1. Address and Port Mapping

When an internal endpoint opens an outgoing 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 mapping 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 mapping
for new sessions to external endpoints, after establishing a first
mapping 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:

```
                                        E
 +------+                    +------+   x
 |  Y1  |                    |  Y2  |   t
 +--+---+                    +---+--+   e
    | Y1:y1            Y2:y2  |        r
     +----------+    +----------+      n
               |    |                  a
        X1':x1' |    | X2':x2'         l
            +--+---+-+
...........|   NAT  |...............
            +--+---+-+              I
               |    |               n
         X:x |    | X:x             t
            ++---++                 e
            |  X  |                 r
            +-----+                 n
                                    a
                                    l
```
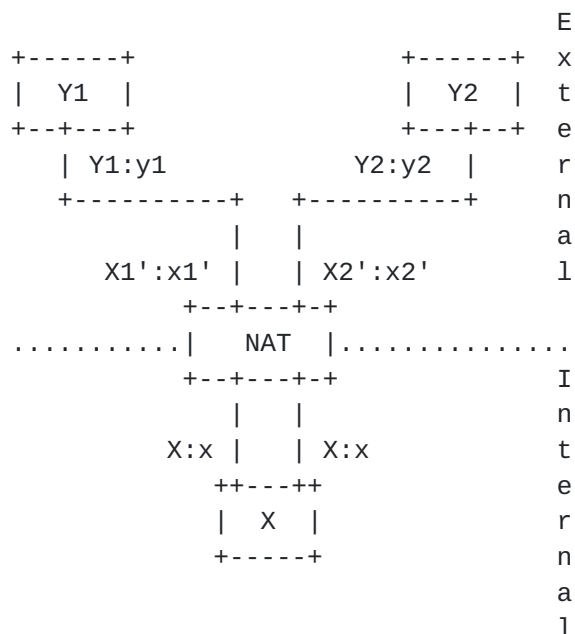
The following address and port mapping behavior are defined:

   Endpoint Independent Mapping:
      The NAT reuses the port mapping for subsequent packets sent
      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.

   Address Dependent Mapping:
      The NAT reuses the port mapping for subsequent packets sent
      from the same internal IP address and port (X:x) to the same
      external IP address, regardless of the external port.
      Specifically, X1':x1' equals X2':x2' if, and only if, Y2 equals
      Y1.

Address and Port Dependent Mapping:
    The NAT reuses the port mapping for subsequent packets sent
    from the same internal IP address and port (X:x) to the same
    external IP address and port while the mapping is still active.
    Specifically, X1':x1' equals X2':x2' if, and only if, Y2:y2
    equals Y1:y1.

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.

REQ-1: A NAT MUST have an "Endpoint Independent Mapping" behavior.

Justification: 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.

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 mapping in an arbitrary fashion
(i.e. randomly): one internal IP address could have multiple external
IP address mappings active at the same time for different sessions.
These NATs have an "IP address pooling" behavior of "Arbitrary".
Some large Enterprise NATs use an IP address pooling behavior of
"Arbitrary" as a means of hiding the IP address assigned to specific
endpoints by making their assignment less predictable. Other NATs
use the same external IP address mapping for all sessions associated
with the same internal IP address. These NATs have an "IP address
pooling" behavior of "Paired." NATs that use an "IP address pooling"
behavior of "arbitrary" can cause issues for applications that use
multiple ports from the same endpoint but do not negotiate IP
addresses individually (e.g., some applications using RTP and RTCP).

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

Justification: This will allow applications that use multiple ports
    originating from the same internal IP address to also have the
    same external IP address. This is to avoid breaking peer-to-peer
    applications that are not capable of negotiating the IP address
    for RTP and the IP address for RTCP separately. As such it is
    envisioned that this requirement will become less important as
    applications become NAT-friendlier with time. The main reason why

this requirement is here is that in a peer-to-peer application,
you are subject to the other peer's mistake. In particular, in
the context of SIP, if my application supports the extensions
defined in RFC 3605 [16] for indicating RTP and RTCP addresses and
ports separately, but the other peer does not, there may still be
breakage in the form of the stream losing RTCP packets. This
requirement will avoid the loss of RTP in this context, although
the loss of RTCP may be inevitable in this particular example. It
is also worth noting that RFC 3605 is unfortunately not a
mandatory part of SIP (RFC 3261). This requirement will therefore
address a particularly nasty problem that will prevail for a
significant period of time.

4.2. Port Assignment

4.2.1. Port Assignment Behavior

This section uses the following diagram for reference.

```
                                         E
    +-------+                +-------+   x
    |  Y1   |                |  Y2   |   t
    +---+---+                +---+---+   e
        | Y1:y1          Y2:y2  |        r
      +---------+   +---------+          n
              |   |                      a
        X1':x1' |   | X2':x2'            l
            +--+---+--+
    ..........|   NAT   |..............
            +--+---+--+                 I
              |   |                      n
      +---------+   +---------+          t
      | X1:x1          X2':x2 |          e
    +---+---+                +---+---+   r
    |  X1   |                |  X2   |   n
    +-------+                +-------+   a
                                         l
```
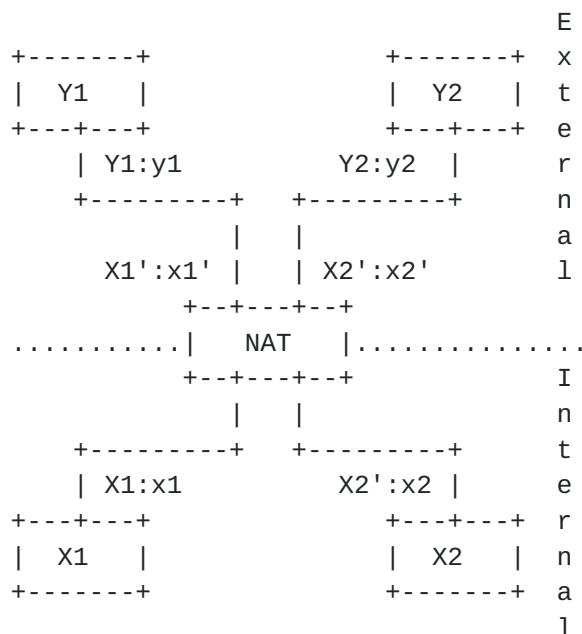
Some NATs attempt to preserve the port number used internally when
assigning a mapping to an external IP address and port (e.g.,
x=x1=x2=x1'=x2', or more succinctly, a mapping of X:x to X':x). A
basic NAT, for example, will preserve the same port and will assign a
different IP address from a pool of 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 Network Address and Port
Translator (NAPT) mode (i.e., X'=X1'=X2' and x=x1=x2 but x1'!=x2', or

a mapping of X1:x to X':x1' and X2:x to X':x2'). This port
assignment behavior 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".

Some NATs use "Port overloading", i.e. they always use port
preservation even in the case of collision (i.e., X'=X1'=X2' and
x=x1=x2=x1'=x2', or a mapping of X1:x to X':x, and X2:x to X':x).
These NATs rely on the source of the response from the external
endpoint (Y1:y1, Y2:y2) 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.

Most applications fail in some cases with "Port overloading". It is
clear that "Port overloading" behavior will result in many problems.
For example it will fail if two internal endpoints try to reach the
same external destination, e.g., a SIP proxy server used by both
endpoints.

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. 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 dynamic range first or
possibly avoid the actual registered ports in the registered range.
Other NATs preserve the port range if it is in the well-known range.

REQ-3: A NAT MUST NOT have a "Port assignment" behavior of "Port
   overloading".
   a) If the host's source port was in the range 1-1023, it is
      RECOMMENDED the NAT's source port be in the same range. If the
      host's source port was in the range 1024-65535, it is
      RECOMMENDED that the NAT's source port be in that range.

Justification: This requirement 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. NATs that
   implement port preservation have to deal with conflicts on ports,
   and the multiple code paths this introduces often result in
   nondeterministic behavior. However, it should be understood that
   when a port is randomly assigned, it may just randomly happen to
   be assigned the same port. Applications must therefore be able to
   deal with both port preservation and no port preservation.
   a) Certain applications expect the source UDP port to be in the
       well-known range. See RFC 2623 for an example.

4.2.2. Port Parity

   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 [14] rule that RTP use
   even ports, and RTCP use odd ports. RFC 3550 allows any port numbers
   to be used for RTP and RTCP if the two numbers are specified
   separately, for example using RFC 3605 [16]. However, some
   implementations do not include RFC 3605 and do not recognize when the
   peer has specified the RTCP port separately using RFC 3605. If such
   an implementation receives an odd RTP port number from the peer
   (perhaps after having been translated by a NAT), and then follows the
   RFC 3550 rule to change the RTP port to the next lower even number,
   this would obviously result in the loss of RTP. NAT-friendly
   application aspects are outside the scope of this document. It is
   expected that this issue will fade away with time, as implementations
   improve. Preserving the port parity allows for supporting
   communication with peers that do not support explicit specification
   of both RTP and RTCP port numbers.

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

   Justification: This is to avoid breaking peer-to-peer applications
      which do not explicitly and separately specify RTP and RTCP port
      numbers and which follow the RFC 3550 rule to decrement an odd RTP
      port to make it even. The same considerations as per the IP
      address pooling requirement apply.

4.2.3. Port Contiguity

   Some NATs attempt to preserve the port contiguity rule of RTCP=RTP+1.
   These NATs do things like sequential assignment or port reservation.
   Sequential port assignment assumes that the application will open a
   mapping for RTP first and then open a mapping for RTCP. It is not
   practical to enforce this requirement on all applications.

Furthermore, there is a glaring problem if many applications (or endpoints) are trying to open mapping simultaneously. Port preservation is also problematic since it is wasteful, especially considering that a NAT cannot reliably distinguish between RTP over UDP and other UDP packets where there is no contiguity rule. For those reasons, it would be too complex to attempt to preserve the contiguity rule by suggesting specific NAT behavior, and it would certainly break the deterministic behavior rule.

In order to support both RTP and RTCP, it will therefore be necessary that applications follow rules to negotiate RTP and RTCP separately, and account for the very real possibility that the RTCP=RTP+1 rule will be broken. As this is an application requirement, it is outside of the scope of this document.

4.3. Mapping Refresh

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

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

REQ-5: A NAT UDP mapping timer MUST NOT expire in less than 2 minutes, unless REQ-5a applies.
   a) A NAT MAY have UDP mapping timers that have much shorter timers, but only for specific ports in the well-known port range (i.e., ports 0-1023) where the IANA- registered protocol is strictly a request/response protocol, such as for example DNS over UDP/53.
   b) The value of the NAT UDP mapping timer MAY be configurable.
   c) A default value of 5 minutes for the NAT UDP mapping timer is RECOMMENDED.

Justification: This requirement is to ensure that the timeout is long enough to avoid too frequent timer refresh packets.
   a) Some UDP protocols using UDP use very short-lived connections. There can be very many such connections; keeping them all in a connections table could cause considerable load on the NAT. Having shorter timers for these specific applications is therefore an optimization technique. It is important that the shorter timers applied to specific protocols be used sparingly, and only for protocols using well-known destination port that are known to have a shorter timer, and that are known not to be used by any applications for other purposes.

       b) Configuration is desirable for adapting to specific networks
          and troubleshooting.
       c) This default is to avoid too frequent timer refresh packets.

   Some NATs keep the mapping 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 a NAT Outbound
   refresh behavior of "True".

   Some NATs keep the mapping 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 a NAT Inbound Refresh Behavior of "True".

   Some NATs keep the mapping active on both, in which case both
   properties are "True".

   REQ-6: The NAT mapping Refresh Direction MUST have a "NAT Outbound
      refresh behavior" of "True".
      a) The NAT mapping Refresh Direction MAY have a "NAT Inbound
         refresh behavior" of "True".

   Justification: Outbound refresh is necessary for allowing the client
      to keep the mapping alive.
      a) Inbound refresh may be useful for applications with no outgoing
         UDP traffic. However, allowing inbound refresh may allow an
         application to keep a mapping alive indefinitely. This may be
         a security risk. Also, if the process is repeated with
         different ports, over time it could use up all the ports on the
         NAT.

4.4. Conflicting Internal and External IP Address Spaces

   Many NATs, particularly consumer-level devices designed to be
   deployed by nontechnical users, routinely obtain their external IP
   address, default router, and other IP configuration information for
   their external interface dynamically from an external network such as
   an upstream ISP. The NAT in turn automatically sets up its own
   internal subnet in one of the private IP address spaces assigned to
   this purpose in RFC 1918 [7], typically providing dynamic IP
   configuration services for hosts on this internal network.

   Auto-configuration of NATs and private networks can be problematic,
   however, if the NAT's external network is also in RFC 1918 private
   address space. In a common scenario, an ISP places its customers
   behind a NAT and hands out private RFC 1918 addresses to them. Some
   of these customers in turn deploy consumer-level NATs, which in
   effect act as "second-level" NATs, multiplexing their own private RFC
   1918 IP subnets onto the single RFC 1918 IP address provided by the

ISP. There is no inherent guarantee in this case that the ISP's
"intermediate" privately-addressed network and the customer's
internal privately-addressed network will not use numerically
identical or overlapping RFC 1918 IP subnets. Customers of consumer-
level NATs further cannot be expected to have the technical knowledge
to prevent this scenario from occurring by manually configuring their
internal network with non-conflicting RFC 1918 subnets.

NAT vendors need to design their NATs to ensure that they function
correctly and robustly even in such problematic scenarios. One
possible solution is for the NAT to ensure that whenever its external
link is configured with an RFC 1918 private IP address, the NAT
automatically selects a different, non-conflicting RFC 1918 IP subnet
for its internal network. A disadvantage of this solution is that if
the NAT's external interface is dynamically configured or re-
configured after its internal network is already in use, then the NAT
may have to renumber its entire internal network dynamically if it
detects a conflict.

An alternative solution is for the NAT to be designed so that it can
translate and forward traffic correctly even when its external and
internal interfaces are configured with numerically overlapping IP
subnets. In this scenario, for example, if the NAT's external
interface has been assigned an IP address P in RFC 1918 space, then
there might also be an internal node I having the same RFC 1918
private IP address P. An IP packet with destination address P on the
external network is directed at the NAT, whereas an IP packet with
the same destination address P on the internal network is directed at
node I. The NAT therefore needs to maintain a clear operational
distinction between "external IP addresses" and "internal IP
addresses" to avoid confusing internal node I with its own external
interface. In general, the NAT needs to allow all internal nodes
(including I) to communicate with all external nodes having public
(non-RFC 1918) IP addresses or having private IP addresses that do
not conflict with the addresses used by its internal network.

REQ-7: A NAT device whose external IP interface can be configured
   dynamically MUST either (1) automatically ensure that its internal
   network uses IP addresses that do not conflict with its external
   network, or (2) be able to translate and forward traffic between
   all internal nodes and all external nodes whose IP addresses do
   not numerically conflict with the internal network.

Justification: If a NAT's external and internal interfaces are
   configured with overlapping IP subnets, then there is of course no
   way for an internal host with RFC 1918 IP address Q to initiate a
   direct communication session to an external node having the same
   RFC 1918 address Q, or to other external nodes with IP addresses

that numerically conflict with the internal subnet. Such nodes
can still open communication sessions indirectly via NAT traversal
techniques, however, with the help of a third-party server such as
a STUN server having a public, non-RFC 1918 IP address. In this
case, nodes with conflicting private RFC 1918 addresses on
opposite sides of the second-level NAT can communicate with each
other via their respective temporary public endpoints on the main
Internet, as long as their common first-level NAT (e.g., the
upstream ISP's NAT) supports hairpinning behavior as described in
Section 6.

5. Filtering Behavior

This section describes various filtering behaviors observed in NATs.

When an internal endpoint opens an outgoing session through a NAT,
the NAT assigns a filtering rule for the mapping 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.

Endpoint Independent Filtering:
   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.

Address Dependent Filtering:
   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:any 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.

Address and Port Dependent Filtering:
   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.

REQ-8: If application transparency is most important, it is
   RECOMMENDED that a NAT have an "Endpoint independent filtering"
   behavior. If a more stringent filtering behavior is most
   important, it is RECOMMENDED that a NAT have an "Address dependent
   filtering" behavior.
   a) The filtering behavior MAY be an option configurable by the
      administrator of the NAT.

Justification: The recommendation to use Endpoint Independent
   Filtering is aimed at maximizing application transparency, in
   particular for applications that receive media simultaneously from
   multiple locations (e.g., gaming), or applications that use
   rendezvous techniques. However, it is also possible that in some
   circumstances, it may be preferable to have a more stringent
   filtering behavior. Filtering independently of the external
   endpoint is not as secure: an unauthorized packet could get
   through a specific port while the port was kept open if it was
   lucky enough to find the port open. 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 policy could interfere
   with applications that expect to receive UDP packets on more than
   one UDP port. Using Endpoint Independent Filtering or Address
   Dependent Filtering instead of Address and Port Dependent
   Filtering on a NAT (say NAT-A) also has benefits when the other
   endpoint is behind a non-BEHAVE compliant NAT (say NAT-B) that
   does not support REQ-1. When the endpoints use ICE, if NAT-A uses
   Address and Port Dependent Filtering, connectivity will require a
   Media Relay. However, if NAT-A uses Endpoint Independent
   Filtering or Address Dependent Filtering, ICE will ultimately find
   connectivity without requiring a Media Relay. Having the
   filtering behavior being an option configurable by the
   administrator of the NAT ensures that a NAT can be used in the
   widest variety of deployment scenarios.

6. **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.

```
       NAT
    +----+ from X1:x1 to X2':x2'    +-----+ X1':x1'
    | X1 |>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>--+---
    +----+                          |  v  |
                                    |  v  |
                                    |  v  |
                                    |  v  |
    +----+ from X1':x1' to X2:x2    |  v  | X2':x2'
    | X2 |<<<<<<<<<<<<<<<<<<<<<<<<<<<<--+---
    +----+                          +-----+
```

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 mapping to an internal address
X2:x2, back to that internal address X2:x2. Note that typically X1'
is the same as 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 implementations
that expect an external IP address and port.

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

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

7. Application Level Gateways

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

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 or protocols that try to be
NAT-aware and must therefore be used with extreme caution.

REQ-10: If a NAT includes ALGs that affect UDP, it is RECOMMENDED
   that all of those ALGs be disabled by default.
   a) If a NAT includes ALGs, it is RECOMMENDED that the NAT allow
      the NAT administrator to enable or disable each ALG separately.

Justification: NAT ALGs may interfere with UNSAF methods.
   a) This requirement allows the user to enable those ALGs that are
      necessary to aid in the operation of some applications without
      enabling ALGs which interfere with the operation of other
      applications.

8. Deterministic Properties

The classification of NATs 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 mapping
characteristics and the External Filter characteristics change
between the X1:x and the X2:x mapping. To make matters worse, there
are NATs where the behavior may be the same on the X1:x and X2:x
mappings but different on the third X3:x mapping.

Another example is that some NATs have an "Endpoint Independent
Mapping" combined with "Port Overloading" as long as two endpoints
are not establishing sessions to the same external direction, but
then switch their behavior to "Address and Port Dependent Mapping"
without "Port Preservation" upon detection of these conflicting
sessions establishments.

Any NAT that changes the NAT Mapping or the Filtering behavior
without configuration changes, at any point in time 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 mapping. The NAT mapping 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 it is
certainly possible that they exist.

REQ-11: A NAT MUST have deterministic behavior, i.e., it MUST NOT
    change the NAT translation (Section 4) or the Filtering
    (Section 5) Behavior at any point in time or under any particular
    conditions.

Justification: 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.

9. ICMP Destination Unreachable Behavior

When a NAT sends a packet towards a host on the other side of the
NAT, an ICMP message may be sent in response to that packet. That
ICMP message may be sent by the destination host or by any router
along the network path. The NAT's default configuration SHOULD NOT
filter ICMP messages based on their source IP address. Such ICMP
messages SHOULD be rewritten by the NAT (specifically the IP headers
and the ICMP payload) and forwarded to the appropriate internal or
external host. The NAT needs to perform this function for as long as
the UDP mapping is active. Receipt of any sort of ICMP message MUST
NOT destroy the NAT mapping. A NAT which performs the functions
described in the paragraph above is referred to as "support ICMP
Processing".

There is no significant security advantage to blocking ICMP
Destination Unreachable packets. Additionally, blocking ICMP
Destination Unreachable packets can interfere with application
failover, UDP Path MTU Discovery (see RFC1191 [3] and RFC1435 [4]),
and traceroute. Blocking any ICMP message is discouraged, and
blocking ICMP Destination Unreachable is strongly discouraged.

REQ-12: Receipt of any sort of ICMP message MUST NOT terminate the
    NAT mapping.
      a) The NAT's default configuration SHOULD NOT filter ICMP messages
         based on their source IP address.
      b) It is RECOMMENDED that a NAT support ICMP Destination
         Unreachable messages.

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

10. Fragmentation of Outgoing Packets

When the MTU of the adjacent link is too small, fragmentation of
packets going from the internal side to the external side of the NAT
may occur. 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, or for
other reasons.

It is worth nothing that many IP stacks do not use Path MTU Discovery
with UDP packets.

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

REQ-13: If the packet received on an internal IP address has DF=1,
    the NAT SHOULD send back an ICMP message "fragmentation needed and
    DF set" message to the host as described in RFC 792 [2].
      a) If the packet has DF=0, the NAT SHOULD fragment the packet and
         send the fragments, in order.

Justification: This is as per RFC 792.
      a) This is the same function a router performs in a similar
         situation RFC 1812 [5].

11. Receiving Fragmented Packets

For a variety of reasons, a NAT may receive a fragmented packet. The
IP packet containing the header could arrive in any fragment
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 fragments in order (that is,
with the header in the first packet) and forwarding each of the
fragments to the internal host is described as "Received Fragments

Ordered".

A NAT that is capable of receiving 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".
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".

REQ-14: A NAT MUST support receiving in order and out of order
   fragments, so it MUST have "Received Fragment Out of Order"
   behavior.
   a) A NAT's out of order fragment processing mechanism MUST be
      designed so that fragmentation-based DoS attacks do not
      compromise the NAT's ability to process in-order and
      unfragmented IP packets.

Justification: See Security Considerations.

12. Requirements

The requirements in this section are aimed at minimizing the
complications caused by NATs to applications such as realtime
communications and online gaming. The requirements listed earlier in
the document are consolidated here into a single section.

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 [18].

A NAT that supports all of the mandatory requirements of this
specification (i.e., the "MUST"), is "compliant with this
specification." A NAT that supports all of the requirements of this
specification (i.e., including the "RECOMMENDED") is "fully compliant
with all the mandatory and recommended requirements of this
specification."

REQ-1: A NAT MUST have an "Endpoint Independent Mapping" behavior.
REQ-2: It is RECOMMENDED that a NAT have an "IP address pooling"
   behavior of "Paired". Note that this requirement is not
   applicable to NATs that do not support IP address pooling.

REQ-3: A NAT MUST NOT have a "Port assignment" behavior of "Port
   overloading".
   a) If the host's source port was in the range 1-1023, it is
      RECOMMENDED the NAT's source port be in the same range. If the
      host's source port was in the range 1024-65535, it is
      RECOMMENDED that the NAT's source port be in that range.
REQ-4: It is RECOMMENDED that a NAT have a "Port parity preservation"
   behavior of "Yes".
REQ-5: A NAT UDP mapping timer MUST NOT expire in less than 2
   minutes, unless REQ-5a applies.
   a) A NAT MAY have UDP mapping timers that have much shorter
      timers, but only for specific ports in the well-known port
      range (i.e., ports 0-1023) where the IANA- registered protocol
      is strictly a request/response protocol, such as for example
      DNS over UDP/53.
   b) The value of the NAT UDP mapping timer MAY be configurable.
   c) A default value of 5 minutes for the NAT UDP mapping timer is
      RECOMMENDED.
REQ-6: The NAT mapping Refresh Direction MUST have a "NAT Outbound
   refresh behavior" of "True".
   a) The NAT mapping Refresh Direction MAY have a "NAT Inbound
      refresh behavior" of "True".
REQ-7 A NAT device whose external IP interface can be configured
   dynamically MUST either (1) Automatically ensure that its internal
   network uses IP addresses that do not conflict with its external
   network, or (2) Be able to translate and forward traffic between
   all internal nodes and all external nodes whose IP addresses do
   not numerically conflict with the internal network.
REQ-8: If application transparency is most important, it is
   RECOMMENDED that a NAT have "Endpoint independent filtering"
   behavior. If a more stringent filtering behavior is most
   important, it is RECOMMENDED that a NAT have "Address dependent
   filtering" behavior.
   a) The filtering behavior MAY be an option configurable by the
      administrator of the NAT.
REQ-9: A NAT MUST support "Hairpinning".
   a) A NAT Hairpinning behavior MUST be "External source IP address
      and port".
REQ-10: If a NAT includes ALGs that affect UDP, it is RECOMMENDED
   that all of those ALGs be disabled by default.
   a) If a NAT includes ALGs, it is RECOMMENDED that the NAT allow
      the NAT administrator to enable or disable each ALG separately.
REQ-11: A NAT MUST have deterministic behavior, i.e., it MUST NOT
   change the NAT translation (Section 4) or the Filtering
   (Section 5) Behavior at any point in time or under any particular
   conditions.

REQ-12: Receipt of any sort of ICMP message MUST NOT terminate the
   NAT mapping.
   a) The NAT's default configuration SHOULD NOT filter ICMP messages
      based on their source IP address.
   b) It is RECOMMENDED that a NAT support ICMP Destination
      Unreachable messages.
REQ-13 If the packet received on an internal IP address has DF=1, the
   NAT SHOULD send back an ICMP message "fragmentation needed and DF
   set" message to the host as described in RFC 792 [2].
   a) If the packet has DF=0, the NAT SHOULD fragment the packet and
      send the fragments, in order.
REQ-14: A NAT MUST support receiving in order and out of order
   fragments, so it MUST have "Received Fragment Out of Order"
   behavior.
   a) A NAT's out of order fragment processing mechanism MUST be
      designed so that fragmentation-based DoS attacks do not
      compromise the NAT's ability to process in-order and
      unfragmented IP packets.

13. 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 mapping 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 mapping 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 mappings
   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 to 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 mapping 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 mapping
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 if done incorrectly. 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 [11] and RFC 1858 [6].

14. IANA Considerations

There are no IANA considerations.

15. 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 [15].

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 [19] 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 [19] provide a reference model for what any UNSAF proposal might encounter in deployed NATs.

16. Acknowledgments

The editor would like to acknowledge Bryan Ford, Pyda Srisuresh and Dan Kegel for the their multiple contributions on peer-to-peer communications across a NAT. Dan Wing contributed substantial text on IP fragmentation and ICMP behavior. Thanks to Rohan Mahy, Jonathan Rosenberg, Mary Barnes, Melinda Shore, Lyndsay Campbell, Geoff Huston, Jiri Kuthan, Harald Welte, Steve Casner, Robert Sanders, Spencer Dawkins, Saikat Guha, Christian Huitema, Yutaka Takeda and Paul Hoffman for their contributions.

17. References

17.1. Normative References

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

17.2. Informational References

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

[3]   Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191,

November 1990.

[4]   Knowles, S., "IESG Advice from Experience with Path MTU
        Discovery", RFC 1435, March 1993.

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

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

[7]   Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E.
        Lear, "Address Allocation for Private Internets", BCP 5,
        RFC 1918, February 1996.

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

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

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

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

[12]   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.

[13]   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.

[14]   Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson,
        "RTP: A Transport Protocol for Real-Time Applications", STD 64,
        RFC 3550, July 2003.

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

[16]   Huitema, C., "Real Time Control Protocol (RTCP) attribute in
        Session Description Protocol (SDP)", RFC 3605, October 2003.

[17]   Rosenberg, J., "Simple Traversal of UDP Through Network Address
        Translators (NAT) (STUN)", draft-ietf-behave-rfc3489bis-03

(work in progress), March 2006.

[18]   Rosenberg, J., "Interactive Connectivity Establishment (ICE): A
       Methodology for Network Address Translator (NAT) Traversal for
       Offer/Answer Protocols", draft-ietf-mmusic-ice-08 (work in
       progress), March 2006.

[19]   Jennings, C., "NAT Classification Test Results",
       draft-jennings-behave-test-results-01 (work in progress),
       July 2005.

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

Authors' Addresses

   Francois Audet (editor)
   Nortel Networks
   4655 Great America Parkway
   Santa Clara, CA 95054
   US

   Phone: +1 408 495 3756
   Email: audet@nortel.com

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

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