NAT Working Group Matt
Holdrege
INTERNET-DRAFT Ascend
Communications
Category: Informational Pyda
Srisuresh
Lucent
Technologies
Expires in six months February
1999

Protocol Complications with the IP Network Address Translator (NAT)

<draft-ietf-nat-protocol-complications-00.txt>

Status of this Memo

Copyright Notice

Abstract:

Many common internet applications can be adversely affected when the communicating end nodes are not in the same routing realm and seek the
assistance of NAT (enroute) to bridge the realms. NAT by itself cannot provide the necessary application/protocol transparency in all cases. Often, a NAT device seeks the assistance of Application Level Gateways (ALGs) to provide the transparency necessary for each application. The purpose of this document is to identify the protocols and applications that cannot function with NAT enroute. The document attempts to

identify
the problem cause and describe known work-arounds and the requirements
on the part of ALGs to make the protocols/applications transparent
with
NAT enroute. It is impossible to capture all the applications and
their
issues with NAT in a single document. This document attempts to
capture
as much information as possible. We hope, the coverage provides
necessary clues for applications not covered by the document.

Introduction:

NAT attempts to provide a transparent routing solution to end hosts
that
need to communicate to disparate routing realms. NAT modifies end node
addresses en-route and maintains state for these updates so that
datagrams pertaining to a session are transparently routed to the
right

end-node in either realm. NAT's fundamental role is to alter the
addresses in the IP header of a packet.

NAT can use much of the same solution set as a Stateful Inspection
firewall. However, the ALG's that complement NAT must also be able to
recompose valid data in the payload, since it must change the address
(and perhaps port) information. This is because the application
running
on a host machine is typically unaware of NAT and may populate
messages
with addressing information as required by the application protocol
and
the addressing information may not be valid on the opposite side of
the
NAT device.

One problem area is when a packet contains significant IP address or port information in the payload of the packet rather than the header. Network applications which use protocols that exhibit this behavior will have problems when a NAT device is in mid-stream. In the next section we will attempt to document standard protocols which have significant address information in the payload of the packet.

Where this document mentions NAT, it is referring to Traditional NAT rather than other NAT techniques.

NOTE the authors wish to make it clear that this work is editorial in nature. Input from the Internet society is requested in order to better cover the range of applications that can be affected by NAT. This is a work in progress.

FTP   REFERENCE: RFC 959

FTP is a TCP based application, used to reliably transfer files between two hosts.

FTP is initiated by a client accessing a well-known port number 21 on the FTP server. This is called the FTP control session. Often, an additional data session accompanies the control session. By default, the data session would be from TCP port 20 on server to the TCP port client used to initiate control session. However, the data session ports may be altered within the FTP control sessions using ASCII encoded PORT and PASV commands and responses.

Say, an FTP client is in a NAT supported private network. An FTP ALG will be required to monitor the FTP control session (for both PORT and PASV modes) to identify the FTP data session port numbers and modify the private address and port number with the externally valid address and port number. In addition, the sequence and acknowledgement numbers, TCP checksum, IP packet length and checksum have to be updated. Consequently the sequence numbers in all subsequent packets for that stream must be adjusted as well as TCP ACK fields and checksums.

Note, the above issue with ASCII encoded address and port can occur with

other applications as well. Changing these numbers can change the size of the overall packet. In rare cases, increasing the size of the packet
could cause it to exceed the MTU of a given transport link. The packet

would then have to be fragmented which could affect performance. Or if the packet has the DF bit set, it would be ICMP rejected and the originating host would then perform Path MTU Discovery. This could also
have an adverse effect on performance.

H.323V1 REFERENCE ITU-T SG16 H.323, Intel white paper, H.323 and Firewalls Dave Chouinard, John Richardson, Milind Khare (with further assistance from Jamie Jason).

H.323 is complex, uses dynamic ports, and includes multiple UDP streams.
Here is a summary of the relevant issues:

An H.323 call is made up of many different simultaneous connections. At
least two of the connections are TCP. For an audio-only conference, there may be up to 4 different UDP 'connections' made.

All connections except one are made to ephemeral (dynamic) ports.

Calls can be initiated from the private as well as the external domain.
For conferencing to be useful, external users need to be able to establish calls directly with internal users' desktop systems.

The addresses and port numbers are exchanged within the data stream of the 'next higher' connection. For example, the port number for the H.245
connection is established within the Q.931 data stream. (This makes it particularly difficult for the ALG, which will be required to modify

the
addresses inside those data streams.) To make matters worse, it is possible in Q.931, for example, to specify that the H.245 connection should be secure (encrypted). If a session is encrypted, it is impossible for the ALG to decipher
 the data stream, unless it has access to the shared key.

Most of the control information is encoded in ASN.1 (only the User-User
Information within Q.931 Protocol Data Units, or PDUs, is ASN.1-encoded
(other parts of each Q.931 PDU are not encoded). For those unfamiliar with ASN.1, suffice it to say that it is a complex encoding scheme, which does not end up with fixed byte offsets for address information. In fact, the same version of the same application connecting to the same
destination may negotiate to include different options, changing the byte offsets.

Below is the protocol exchange for a typical H.323 call between User A and User B. A's IP address is 88.88.88.88 and B's IP address is 99.99.99.99. Note that the Q.931 and H.245 messages are encoded in ASN.1 in the payload of an RTP packet. So to accomplish a connection through a NAT device, an H.323-ALG will be required to examine the packet, decode the ASN.1, and translate the various H.323 control IP addresses.

```
User A                                                  User B
      A establishes connection to B on well-
      known Q.931 port (1720)
```



```
      ------------------------------------------------>
      Q.931 Setup caller address = 88.88.88.88
                  caller port    = 1120
                  callee address = 99.99.99.99
                  callee port    = 1720
```

```
<-----------------------------------------------
Q.931 Alerting
<-----------------------------------------------
Q.931 Connect H.245 address = 99.99.99.99
              H.245 port    = 1092

User A establishes connection to User B at
99.99.99.99, port 1092

<---------------------------------------------->
Several H.245 messages are exchanged (Terminal
Capability Set, Master Slave Determination and
their respective ACKs)

<-----------------------------------------------
H.245 Open Logical Channel, channel = 257
          RTCP address = 99.99.99.99
          RTCP port    = 1093
----------------------------------------------->
H.245 Open Logical Channel Ack, channel = 257
          RTP address = 88.88.88.88
          RTP port    = 2002
          (This is where User A would like RTP
           data sent to)
          RTCP address = 88.88.88.88
          RTCP port    = 2003
----------------------------------------------->
H.245 Open Logical Channel, channel = 257
          RTCP address = 88.88.88.88
          RTCP port    = 2003
<-----------------------------------------------
H.245 Open Logical Channel Ack, channel = 257
          RTP address = 99.99.99.99
          RTP port    = 1092
          (This is where User B would like RTP data
           sent to)
          RTCP address = 99.99.99.99
          RTP port     = 1093
```

Also note that if an H.323 Gateway resided inside a NAT boundary, the ALG would have to be cognizant of the various gateway discovery schemes and adapt to those schemes as well. Or if just the H.323 host/terminal was inside the NAT boundary and tried to register with a Gatekeeper, the IP information in the registration messages would have to be translated by NAT.

RSVP   Reference RFC 2205

RSVP is positioned in the protocol stack at the transport layer, operating On top of IP (either IPv4 or IPv6). However, unlike other transport protocols, RSVP does not transport application data but instead acts like other Internet control protocols (for example, ICMP, IGMP, routing protocols). RSVP messages are sent hop-by-hop between RSVP-capable routers as raw IP datagrams using protocol number 46. It is intended that raw IP datagrams should be used between the end systems and the first (or last) hop router. However, this may not always be possible as not all systems can do raw network I/O. Because of this, it is possible to encapsulate RSVP messages within UDP datagrams for end-system communication. UDP-encapsulated RSVP messages are sent to either port 1698 (if sent by an end system) or port 1699 (if sent by an RSVP-enabled router). For more information concerning UDP encapsulation of RSVP messages, consult Appendix C of RFC 2205.

An RSVP session, a data flow with a particular destination and transport-layer protocol, is defined by:

Destination Address - the destination IP address for the data packets. This may be either a unicast or a multicast address.

Protocol ID - the IP protocol ID (for example, UDP or TCP).

Destination Port - a generalized destination port which is used for demultiplexing at a layer above the IP layer.

NAT devices are presented with unique problems when it comes to supporting RSVP. Two issues are...

1. RSVP message objects may contain IP addresses. The result is that an

RSVP-ALG must be able to replace the IP addresses based upon the direction and type of the message. For example, if an external sender were to send an RSVP Path message to an internal receiver, the RSVP Session will specify the IP address that the external sender believes is the IP address of the internal receiver. However, when the RSVP Path message reaches the NAT device, the RSVP Session must be changed to reflect the IP address that is used internally for the receiver. Similar actions must be taken for all message objects that contain IP addresses.

2. RSVP provides a means, the RSVP Integrity object, to guarantee the integrity of RSVP messages. The problem is that because of the first point, a NAT device must be able to change IP addresses within the RSVP messages. However, when this is done, the RSVP Integrity object is no longer valid as the RSVP message has been changed.

DNS:

Domain Names are an issue for hosts which use local DNS servers behind a NAT device. Such servers return site specific information which may conflict with external domain addresses.

Zone transfers from private routing realm to external realm must be avoided for address assignments that are not static. If primary and

backup name servers are in the same private domain, zone transfer do not cross the realm and DNS_ALG support for zone transfer is not an issue.

CHARACTERISTICS:

A. UDP based protocol.

B. Inverse name lookup queries embed the IP address in ASCII format. For example, a resolver that wanted to find the hostname of an address 198.76.29.1 (externally assigned address of a private realm host) would pursue a query of the form:

QTYPE = PTR, QCLASS= IN, QNAME = 1.29.76.198.IN-ADDR.ARPA

An ALG is required to translate the query while forwarding to a DNS server within private realm, so that the query will appear as follows (replacing the externally assigned address with its private address).

QTYPE = PTR, QCLASS= IN, QNAME = 1.0.0.10.IN-ADDR.ARPA

Clearly, payload length could change when payload is translated.

C. Serial reuse of an address assignment between independent sessions. This requires that the ALG keep the address assignment (between private and external addresses) valid for a pre-configured period of time, past the DNS query.

Ex: DNS queries assume that the address assigned in response to a name lookup is serially reusable by a follow-on application.

D. A single DNS query payload could contain multiple queries at the same time, requiring translation of multiple addresses within a private domain.

CONFIGURATION ISSUES:

DNS name to address mapping for hosts in private domain should be configured on an authorititive name server within the private domain. This server would be accessed by external and internal hosts alike for name resolutions. A DNS ALG would be required to perform address to name conversions on DNS queries and responses.

Alternately, if there isnt a need for a name server within private domain, private domain hosts could simply point to an external name server for external name lookup. No ALG is required when the name server is located in external domain.

WHAT BREAKS: Authoritative name server for public domain access mUst not contain hosts with private IP addresses.

ADDITIONAL INFO: Refer RFC 1034, RFC 1035, DNS-ALG draft

EMAIL: E-Mail programs - sendmail, Eudora, and others.

DESCRIPTION: The e-mail programs listed above operate based on a TCP based SMTP protocol and use a well-known port number 25 to send messages and to listen on incoming messages.

CHARACTERISTICS:

A. SMTP is a TCP based protocol, based on a well known TCP port number 25.

B. In the majority of cases, mail messages do not contain reference to private IP addresses or links to content data via names that are not visible to outside.

Some mail messages do contain IP addresses of the MTAs that relay the message in the "Received: " field. Some mail messages use IP addresses in place of FQDN for debug purposes or due to lack of a DNS record, in "Mail From: " field.

CONFIGURATION ISSUES:

You need to specify a mail server, with a globally assigned IP address to receive mail from external hosts.

Generally speaking, you would want to configure your mail system such that all users specify a single centralized address (such as fooboo@company.com), instead of including individual hosts (such as fooboo@hostA.company.com). The central address must have an MX record specified in the DNS name server accessible by external hosts.

The mail server may be located within or outside private domain. But, the requirement is that the server be assigned a global name and address, accessible by external hosts.

If one or more MTAs were to be located behind NAT in private domain, an
SMTP-ALG will be required to translate the IP address information registered by the MTAs. Typically, the MTAs will be expected to have a static address mapping make the translation valid across realms for long
periods of time.

When mail server is located within private domain, inbound SMTP sessions
must be redirected to the private host from its externally assigned address. No special mapping is required when Mail server is located in external domain.

WHAT BREAKS: You do not have an SMTP-ALG and yet the mail message or headers contains reference to private IP addresses or links to content

data via names that are not visible to the outside. The ability to trace
the mail route may also be hampered or prevented by NAT. This can consequently cause problems when debugging mail problems or tracking down abusive users of mail.

ADDITIONAL INFO: RFC 821.

X-Windows:

DESCRIPTION: These applications are TCP based. However, the client-

server relationship with these applications is reverse compared to
most
other applications. The X-server or Open-windows server is the
display/mouse/keyboard unit (i.e., the one that controls the actual
Windows interface). The clients are the application programs driving
the
Windows interface.

Some machines run multiple X-Windows servers on the same machine. The
first X-windows server is at TCP port 6000. The first Open Windows
server can be at port 6000 or port 2000 (more flexible). We will
refer
X-windows mainly for illustration purposes here.

On a UNIX system, the csh DISPLAY command "setenv DISPLAY
<hostname>:n",
where n>= 0, is used to tell clients to contact X server on <hostname>
on TCP port (6000+n).

A common use of this application is people dialing in to corporate
offices from their X terminals at home.

CHARACTERISTICS:

A. X-Windows is a TCP based protocol, with the server
  servicing TCP ports in the range of 6000 - 6000+n.
  Open-Windows is also a TCP based protocol, with the server
  servicing TCP ports in the range of 6000 - 6000+n or
  2000 - 2000+n.

B. The X-Windows applications are not expected to contain
  reference to private IP addresses or links to content
  data via names that are not visible to the outside. All
  the information required for Client-Server communication
  is in the IP and TCP headers.

CONFIGURATION ISSUES:

When X-Windows server (i.e., the machine that displays the X-Windows
on
its console) runs in a private domain, we need to allow inbound X-
server
access for the X terminals at home. I.e., Users that need to provide
X-
terminal access must have inbound access permissions. This can be
done
statically or dynamically for private hosts.

In case of a NAPT setup, the individual X-Windows ports namely, 6000,
6001, 6002, 6003 and so on till (6000+n) on the external address may

be

statically redirected to different hosts running X-server.

For Example, you could redirect inbound TCP sessions to <External address>:6000 to <private Host A>, sessions to <External Address>:6001 to <private Host B> and so on.

WHAT BREAKS: Accessing more X-servers than are configured.

ADDITIONAL INFO: RFC 1198.

SIP (Session Initiation Protocol)

Description: SIP can run on either TCP or UDP, but by default on the same port; 5060.

When used with UDP, a response to a SIP request does not go to the source port the request came from. Rather, the SIP message contains the port number the reponse should be sent to. SIP makes use of ICMP port unreachable errors in response to request transmissions. Request messages are usually sent on the connected socket. If responses are sent to the source port in the request, each thread handling a request would have to listen on the socket it sent the request on. However, by allowing responses to come to a single port, a single thread can be used for listening instead.

A server may prefer to place the source port of each connected socket in the message. Then each thread can listen for responses separately. Since the port number for a response may not go to the source port of the request, SIP will not normally traverse a NAT and would require a SIP-ALG.

SIP messages carry arbitrary content which is defined by a MIME type. For multimedia sessions, this is usually the Session Description Protocol (SDP RFC 2327). SDP may specify IP addresses or ports to be used for the exchange of multimedia. These may lose significance when traversing a NAT. Thus a SIP-ALG would need the intelligence to decipher
and translate realm-relevant information.

SIP carries URL's in its Contact, To and From fields that specify signalling addresses. These URL's can contain IP addresses or domain names in the host port portion of the URL. These may not be valid once they traverse a NAT.

As an alternative to an SIP-ALG, SIP supports a proxy server which could
co-reside with NAT and function on the globally significant NAT port. Such a proxy would have to a locally specific configuration.

RealAudio

DESCRIPTION: In its default mode, clients (say, in a private domain)
access TCP port 7070 to initiate conversation with a real-audio server (say, located an external domain) and to exchange control messages

during playback (ex: pausing or stopping the audio stream).

The actual audio traffic is carried on incoming UDP based packets (originated from the server) directed to ports in the range of 6970-7170.

CHARACTERISTICS:

A. Real Audio has a TCP control session in one direction directed to a well-known port (7070) and the UDP based audio session in the opposite direction.

B. Audio session parameters are embedded in the TCP control session as byte stream(?)

CONFIGURATION

You could have an ALG examine the TCP traffic to determine the audio session parameters and selectively enable inbound UDP sessions for the ports agreed upon in the TCP control session. Alternately, the ALG could simply redirect all inbound UDP sessions directed to ports 6970-7170 to the client address in the private domain.

For bi-Directional NAT, you will not need an ALG. Bi-directional NAT could simply treat each of the TCP and UDP sessions as 2 unrelated sessions and simply perform IP and TCP/UDP header level translations.

WHAT BREAKS:

ADDITIONAL INFO: http://www.real.com/firewall/packetfil.html

Activision Games

DESCRIPTION: The goal of Activision Games is to work transparently through traditional NAT devices. As such, the protocol described is intended to be NAT friendly so game players within a private domain can
play with other players in the same domain or external domain.

All peers are somehow informed of each others' public and private addresses, and each client opens up symmetrical direct connections to each other and use whichever address (private or external) works first.

Now, the clients can have a session directly with other clients directly
(or) they can have session with other clients via the gaming server.

CHARACTERISTICS:

A. Activision gaming protocol is proprietary and is based on UDP. The server uses UDP port no. 21157.

B. The protocol is designed with keeping NAT and NAPT in mind. The game players can be within the same private domain, in a combination of multiple private domains and external domain.

C. The key is to allow the reuse of the tuple of the same (global address, assigned UDP port) for initial connection to the game server (helper) and the subsequent connection to the client. A game player is recognized by one of (private address, UDP port) or (Assigned global address, assigned UDP port) by all other peer players. So, the binding between tuples should remain unchanged so long as the gaming player is in session with one or multiple other players.

CONFIGURATION ISSUES:

Opening a connection to a game server in external realm from a private host is no problem. All NAT would have to do is provide routing transparency.

But, an NAPT configuration MUST allow multiple simultaneous UDP connections on the same assigned global address/port.

ADDITIONAL INFO:

http://www.csn.tu-chemnitz.de/HyperNews/get/linux-ip-nat/97.html
http://newjersey1.activision.com/anet2
http://california3.activision.com/anet2

ROUTING UPDATES:

Routing advertisement varies considerably based on the NAT flavor in use. A traditional-NAT and bi-directional-NAT may advertise external routes to the private realm, yet not translated. However, a Twice-NAT device must translate external routes (into their private realm address blocks), if it chooses to advertise those routes into private realm.

All flavors of NAT must refrain from advertising private realm routes into external realms. Instead, every NAT device must advertise (or be made apparent through static configuration of neighboring routers or some other means) the external address block it uses for mapping private realm addresses.

SECURITY:

Another class of problems with NAT is end-to-end security of packets. The IPsec AH standard [RFC 1826] is explicitly intended to detect what NAT is good at. That is altering the header of the packet. So when NAT alters the address information in the header of the packet, the destination host receives the altered packet and begins digesting the AH message. The AH routines at this host will invalidate the packet since the contents of the headers have been altered. Depending on the configuration of the end host, the packet could be simply dropped, or

higher layer security activities could be started.

Other IPsec protocols with NAT complications:

ESP: Protects/obscures the packet contents (which would need to be visible for NATing some protocols).

IKE: Potentially passes IP addresses during both Main, Aggressive and Quick Modes. In order for a negotiation to correctly pass through a NAT, these payloads would need to be modified. However, these payloads are often protected by hash or obscured by encryption.

Authors Addresses:

Matt Holdrege
Ascend Communications, Inc.
One Ascend Plaza
1701 Harbor Bay Parkway
Alameda, CA 94502
Voice: (510) 769-6001
EMail: matt@ascend.com

Pyda Srisuresh
Lucent technologies
4464 Willow Road
Pleasanton, CA 94588-8519
U.S.A.
Voice: (925) 737-2153
EMail: suresh@ra.lucent.com