Network Working Group P. Srisuresh
Request for Comments: 3022 Jasmine Networks
Obsoletes: 1631 K. Egevang
Category: Informational Intel Corporation
January 2001

Traditional IP Network Address Translator (Traditional NAT)

Status of this Memo

Copyright Notice

Preface

The NAT operation described in this document extends address translation introduced in RFC 1631 and includes a new type of network address and TCP/UDP port translation. In addition, this document corrects the Checksum adjustment algorithm published in RFC 1631 and attempts to discuss NAT operation and limitations in detail.

Abstract

Basic Network Address Translation or Basic NAT is a method by which IP addresses are mapped from one group to another, transparent to end users. Network Address Port Translation, or NAPT is a method by which many network addresses and their TCP/UDP (Transmission Control Protocol/User Datagram Protocol) ports are translated into a single network address and its TCP/UDP ports. Together, these two operations, referred to as traditional NAT, provide a mechanism to connect a realm with private addresses to an external realm with globally unique registered addresses.

1. Introduction

The need for IP Address translation arises when a network's internal IP addresses cannot be used outside the network either for privacy reasons or because they are invalid for use outside the network.

Network topology outside a local domain can change in many ways. Customers may change providers, company backbones may be reorganized, or providers may merge or split. Whenever external topology changes with time, address assignment for nodes within the local domain must also change to reflect the external changes. Changes of this type can be hidden from users within the domain by centralizing changes to a single address translation router.

Basic Address translation would (in many cases, except as noted in [NAT-TERM] and section 6 of this document) allow hosts in a private network to transparently access the external network and enable access to selective local hosts from the outside. Organizations with a network setup predominantly for internal use, with a need for occasional external access are good candidates for this scheme.

Many Small Office, Home Office (SOHO) users and telecommuting employees have multiple Network nodes in their office, running TCP/UDP applications, but have a single IP address assigned to their remote access router by their service provider to access remote networks. This ever increasing community of remote access users would be benefited by NAPT, which would permit multiple nodes in a local network to simultaneously access remote networks using the single IP address assigned to their router.

There are limitations to using the translation method. It is mandatory that all requests and responses pertaining to a session be routed via the same NAT router. One way to ascertain this would be to have NAT based on a border router that is unique to a stub domain, where all IP packets are either originated from the domain or destined to the domain. There are other ways to ensure this with multiple NAT devices. For example, a private domain could have two distinct exit points to different providers and the session flow from the hosts in a private network could traverse through whichever NAT device has the best metric for an external host. When one of the NAT routers fail, the other could route traffic for all the connections. There is however a caveat with this approach, in that, rerouted flows could fail at the time of switchover to the new NAT router. A way to overcome this potential problem is that the routers share the same NAT configuration and exchange state information to ensure a fail-safe backup for each other.

Address translation is application independent and often accompanied by application specific gateways (ALGs) to perform payload monitoring and alterations. FTP is the most popular ALG resident on NAT devices. Applications requiring ALG intervention must not have their payload encoded, as doing that would effectively disables the ALG, unless the ALG has the key to decrypt the payload.

This solution has the disadvantage of taking away the end-to-end significance of an IP address, and making up for it with increased state in the network. As a result, end-to-end IP network level security assured by IPSec cannot be assumed to end hosts, with a NAT device enroute. The advantage of this approach however is that it can be installed without changes to hosts or routers.

Definition of terms such as "Address Realm", "Transparent Routing", "TU Ports", "ALG" and others, used throughout the document, may be found in [NAT-TERM].

2. Overview of traditional NAT

The Address Translation operation presented in this document is referred to as "Traditional NAT". There are other variations of NAT that will not be explored in this document. Traditional NAT would allow hosts within a private network to transparently access hosts in the external network, in most cases. In a traditional NAT, sessions are uni-directional, outbound from the private network. Sessions in the opposite direction may be allowed on an exceptional basis using static address maps for pre-selected hosts. Basic NAT and NAPT are two variations of traditional NAT, in that translation in Basic NAT is limited to IP addresses alone, whereas translation in NAPT is extended to include IP address and Transport identifier (such as TCP/UDP port or ICMP query ID).

Unless mentioned otherwise, Address Translation or NAT throughout this document will pertain to traditional NAT, namely Basic NAT as well as NAPT. Only the stub border routers as described in figure 1 below may be configured to perform address translation.

```
     \ | /                 .                             /
+---------------+  WAN     .           +-----------------+/
|Regional Router|----------------------|Stub Router w/NAT|---
+---------------+          .           +-----------------+\
                           .                      |        \
                           .                      |  LAN
                           .               ---------------
                     Stub border
```

Figure 1: Traditional NAT Configuration

2.1 Overview of Basic NAT

Basic NAT operation is as follows. A stub domain with a set of private network addresses could be enabled to communicate with external network by dynamically mapping the set of private addresses to a set of globally valid network addresses. If the number of local nodes are less than or equal to addresses in the global set, each local address is guaranteed a global address to map to. Otherwise, nodes allowed to have simultaneous access to external network are limited by the number of addresses in global set. Individual local addresses may be statically mapped to specific global addresses to ensure guaranteed access to the outside or to allow access to the local host from external hosts via a fixed public address. Multiple simultaneous sessions may be initiated from a local node, using the same address mapping.

Addresses inside a stub domain are local to that domain and not valid outside the domain. Thus, addresses inside a stub domain can be reused by any other stub domain. For instance, a single Class A address could be used by many stub domains. At each exit point between a stub domain and backbone, NAT is installed. If there is more than one exit point it is of great importance that each NAT has the same translation table.

For instance, in the example of figure 2, both stubs A and B internally use class A private address block 10.0.0.0/8 [RFC 1918]. Stub A’s NAT is assigned the class C address block 198.76.29.0/24, and Stub B’s NAT is assigned the class C address block 198.76.28.0/24. The class C addresses are globally unique no other NAT boxes can use them.

```
                                    \ | /
                                  +---------------+
                                  |Regional Router|
                                  +---------------+
                                WAN |           | WAN
                                    |           |
                Stub A .............|....   ....|............ Stub B
                                    |           |
                  {s=198.76.29.7,^  |           |  v{s=198.76.29.7,
                   d=198.76.28.4}^  |           |  v d=198.76.28.4}
                    +-----------------+       +-----------------+
                    |Stub Router w/NAT|       |Stub Router w/NAT|
                    +-----------------+       +-----------------+
                          |                         |
                          |  LAN               LAN  |
                    -------------             -------------
                              |                 |
            {s=10.33.96.5, ^  |                 |  v{s=198.76.29.7,
             d=198.76.28.4}^ +--+             +--+ v d=10.81.13.22}
                             |--|             |--|
                            /____\           /____\
                          10.33.96.5       10.81.13.22
```

Figure 2: Basic NAT Operation

When stub A host 10.33.96.5 wishes to send a packet to stub B host 10.81.13.22, it uses the globally unique address 198.76.28.4 as destination, and sends the packet to its primary router. The stub router has a static route for net 198.76.0.0 so the packet is forwarded to the WAN-link. However, NAT translates the source address 10.33.96.5 of the IP header to the globally unique 198.76.29.7 before the packet is forwarded. Likewise, IP packets on the return path go through similar address translations.

Notice that this requires no changes to hosts or routers. For instance, as far as the stub A host is concerned, 198.76.28.4 is the address used by the host in stub B. The address translations are transparent to end hosts in most cases. Of course, this is just a simple example. There are numerous issues to be explored.

2.2. Overview of NAPT

Say, an organization has a private IP network and a WAN link to a service provider. The private network’s stub router is assigned a globally valid address on the WAN link and the remaining nodes in the organization have IP addresses that have only local significance. In such a case, nodes on the private network could be allowed simultaneous access to the external network, using the single registered IP address with the aid of NAPT. NAPT would allow mapping of tuples of the type (local IP addresses, local TU port number) to tuples of the type (registered IP address, assigned TU port number).

This model fits the requirements of most Small Office Home Office (SOHO) groups to access external network using a single service provider assigned IP address. This model could be extended to allow inbound access by statically mapping a local node per each service TU port of the registered IP address.

In the example of figure 3 below, stub A internally uses class A address block 10.0.0.0/8. The stub router’s WAN interface is assigned an IP address 138.76.28.4 by the service provider.

```
                                     \ | /
                                   +-----------------------+
                                   |Service Provider Router|
                                   +-----------------------+
                                 WAN |
                                     |
                 Stub A .............|....
                                     |
         ^{s=138.76.28.4,sport=1024, |  v{s=138.76.29.7, sport = 23,
         ^ d=138.76.29.7,dport=23}   |  v d=138.76.28.4, dport = 1024}
                         +-------------------+
                         |Stub Router w/NAPT|
                         +-------------------+
                           |
                           |  LAN
     --------------------------------------------
       |        ^{s=10.0.0.10,sport=3017, |  v{s=138.76.29.7, sport=23,
       |        ^ d=138.76.29.7,dport=23} |  v d=10.0.0.10, dport=3017}
       |                                  |
      +--+      +--+                    +--+
      |--|      |--|                    |--|
     /____\    /____\                  /____\
   10.0.0.1  10.0.0.2   .....        10.0.0.10
```

Figure 3: Network Address Port Translation (NAPT) Operation

When stub A host 10.0.0.10 sends a telnet packet to host 138.76.29.7, it uses the globally unique address 138.76.29.7 as destination, and sends the packet to it's primary router. The stub router has a static route for the subnet 138.76.0.0/16 so the packet is forwarded to the WAN-link. However, NAPT translates the tuple of source address 10.0.0.10 and source TCP port 3017 in the IP and TCP headers into the globally unique 138.76.28.4 and a uniquely assigned TCP port, say 1024, before the packet is forwarded. Packets on the return path go through similar address and TCP port translations for the target IP address and target TCP port. Once again, notice that this requires no changes to hosts or routers. The translation is completely transparent.

In this setup, only TCP/UDP sessions are allowed and must originate from the local network. However, there are services such as DNS that demand inbound access. There may be other services for which an organization wishes to allow inbound session access. It is possible to statically configure a well known TU port service [RFC 1700] on the stub router to be directed to a specific node in the private network.

In addition to TCP/UDP sessions, ICMP messages, with the exception of REDIRECT message type may also be monitored by NAPT router. ICMP query type packets are translated similar to that of TCP/UDP packets, in that the identifier field in ICMP message header will be uniquely mapped to a query identifier of the registered IP address. The identifier field in ICMP query messages is set by Query sender and returned unchanged in response message from the Query responder. So, the tuple of (Local IP address, local ICMP query identifier) is mapped to a tuple of (registered IP address, assigned ICMP query Identifier) by the NAPT router to uniquely identify ICMP queries of all types from any of the local hosts. Modifications to ICMP error messages are discussed in a later section, as that involves modifications to ICMP payload as well as the IP and ICMP headers.

In NAPT setup, where the registered IP address is the same as the IP address of the stub router WAN interface, the router has to be sure to make distinction between TCP, UDP or ICMP query sessions originated from itself versus those originated from the nodes on local network. All inbound sessions (including TCP, UDP and ICMP query sessions) are assumed to be directed to the NAT router as the end node, unless the target service port is statically mapped to a different node in the local network.

Sessions other than TCP, UDP and ICMP query type are simply not permitted from local nodes, serviced by a NAPT router.

3.0. Translation phases of a session.

The translation phases with traditional NAT are same as described in [NAT-TERM]. The following sub-sections identify items that are specific to traditional NAT.

3.1. Address binding:

With Basic NAT, a private address is bound to an external address, when the first outgoing session is initiated from the private host. Subsequent to that, all other outgoing sessions originating from the same private address will use the same address binding for packet translation.

In the case of NAPT, where many private addresses are mapped to a single globally unique address, the binding would be from the tuple of (private address, private TU port) to the tuple of (assigned address, assigned TU port). As with Basic NAT, this binding is determined when the first outgoing session is initiated by the tuple of (private address, private TU port) on the private host. While not a common practice, it is possible to have an application on private host establish multiple simultaneous sessions originating from the same tuple of (private address, private TU port). In such a case, a single binding for the tuple of (private address, private TU port) may be used for translation of packets pertaining to all sessions originating from the same tuple on a host.

3.2. Address lookup and translation:

After an address binding or (address, TU port) tuple binding in case of NAPT is established, a soft state may be maintained for each of the connections using the binding. Packets belonging to the same session will be subject to session lookup for translation purposes. The exact nature of translation is discussed in the follow-on section.

3.3. Address unbinding:

When the last session based on an address or (address, TU port) tuple binding is terminated, the binding itself may be terminated.

4.0. Packet Translations

Packets pertaining to NAT managed sessions undergo translation in either direction. Individual packet translation issues are covered in detail in the following sub-sections.

4.1. IP, TCP, UDP and ICMP Header Manipulations

In Basic NAT model, the IP header of every packet must be modified. This modification includes IP address (source IP address for outbound packets and destination IP address for inbound packets) and the IP checksum.

For TCP ([TCP]) and UDP ([UDP]) sessions, modifications must include update of checksum in the TCP/UDP headers. This is because TCP/UDP checksum also covers a pseudo header which contains the source and destination IP addresses. As an exception, UDP headers with 0 checksum should not be modified. As for ICMP Query packets ([ICMP]), no further changes in ICMP header are required as the checksum in ICMP header does not cover IP addresses.

In NAPT model, modifications to IP header are similar to that of Basic NAT. For TCP/UDP sessions, modifications must be extended to include translation of TU port (source TU port for outbound packets and destination TU port for inbound packets) in the TCP/UDP header. ICMP header in ICMP Query packets must also be modified to replace the query ID and ICMP header checksum. Private host query ID must be translated into assigned ID on the outbound and the exact reverse on the inbound. ICMP header checksum must be corrected to account for Query ID translation.

4.2. Checksum Adjustment

NAT modifications are per packet based and can be very compute intensive, as they involve one or more checksum modifications in addition to simple field translations. Luckily, we have an algorithm below, which makes checksum adjustment to IP, TCP, UDP and ICMP headers very simple and efficient. Since all these headers use a one's complement sum, it is sufficient to calculate the arithmetic difference between the before-translation and after-translation addresses and add this to the checksum. The algorithm below is applicable only for even offsets (i.e., optr below must be at an even offset from start of header) and even lengths (i.e., olen and nlen below must be even). Sample code (in C) for this is as follows.

```
void checksumadjust(unsigned char *chksum, unsigned char *optr,
int olen, unsigned char *nptr, int nlen)
/* assuming: unsigned char is 8 bits, long is 32 bits.
  - chksum points to the chksum in the packet
  - optr points to the old data in the packet
  - nptr points to the new data in the packet
*/
{
  long x, old, new;
  x=chksum[0]*256+chksum[1];
  x=~x & 0xFFFF;
  while (olen)
  {
      old=optr[0]*256+optr[1]; optr+=2;
      x-=old & 0xffff;
      if (x<=0) { x--; x&=0xffff; }
      olen-=2;
  }
  while (nlen)
  {
      new=nptr[0]*256+nptr[1]; nptr+=2;
      x+=new & 0xffff;
      if (x & 0x10000) { x++; x&=0xffff; }
      nlen-=2;
  }
  x=~x & 0xFFFF;
  chksum[0]=x/256; chksum[1]=x & 0xff;
}
```

4.3. ICMP error packet modifications

Changes to ICMP error message ([ICMP]) will include changes to IP and ICMP headers on the outer layer as well as changes to headers of the packet embedded within the ICMP-error message payload.

In order for NAT to be transparent to end-host, the IP address of the IP header embedded within the payload of ICMP-Error message must be modified, the checksum field of the embedded IP header must be modified, and lastly, the ICMP header checksum must also be modified to reflect changes to payload.

In a NAPT setup, if the IP message embedded within ICMP happens to be a TCP, UDP or ICMP Query packet, you will also need to modify the appropriate TU port number within the TCP/UDP header or the Query Identifier field in the ICMP Query header.

Lastly, the IP header of the ICMP packet must also be modified.

4.4. FTP support

One of the most popular applications, "FTP" ([FTP]) would require an ALG to monitor the control session payload to determine the ensuing data session parameters. FTP ALG is an integral part of most NAT implementations.

The FTP ALG would require a special table to correct the TCP sequence and acknowledge numbers with source port FTP or destination port FTP. The table entries should have source address, destination address, source port, destination port, delta for sequence numbers and a timestamp. New entries are created only when FTP PORT commands or PASV responses are seen. The sequence number delta may be increased or decreased for every FTP PORT command or PASV response. Sequence numbers are incremented on the outbound and acknowledge numbers are decremented on the inbound by this delta.

FTP payload translations are limited to private addresses and their assigned external addresses (encoded as individual octets in ASCII) for Basic NAT. For NAPT setup, however, the translations must be extended to include the TCP port octets (in ASCII) following the address octets.

4.5 DNS support

Considering that sessions in a traditional NAT are predominantly outbound from a private domain, DNS ALG may be obviated from use in conjunction with traditional NAT as follows. DNS server(s) internal to the private domain maintain mapping of names to IP addresses for internal hosts and possibly some external hosts. External DNS servers maintain name mapping for external hosts alone and not for any of the internal hosts. If the private network does not have an internal DNS server, all DNS requests may be directed to external DNS server to find address mapping for the external hosts.

4.6. IP option handling

An IP datagram with any of the IP options Record Route, Strict Source Route or Loose Source Route would involve recording or using IP addresses of intermediate routers. A NAT intermediate router may choose not to support these options or leave the addresses untranslated while processing the options. The result of leaving the addresses untranslated would be that private addresses along the source route are exposed end to end. This should not jeopardize the traversal path of the packet, per se, as each router is supposed to look at the next hop router only.

5. Miscellaneous issues

5.1. Partitioning of Local and Global Addresses

For NAT to operate as described in this document, it is necessary to partition the IP address space into two parts - the private addresses used internal to stub domain, and the globally unique addresses. Any given address must either be a private address or a global address. There is no overlap.

The problem with overlap is the following. Say a host in stub A wished to send packets to a host in stub B, but the global addresses of stub B overlapped the private addressees of stub A. In this case, the routers in stub A would not be able to distinguish the global address of stub B from its own private addresses.

5.2. Private address space recommendation

[RFC 1918] has recommendations on address space allocation for private networks. Internet Assigned Numbers Authority (IANA) has three blocks of IP address space, namely 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 for private internets. In pre-CIDR notation, the first block is nothing but a single class A network number, while the second block is a set of 16 contiguous class B networks, and the third block is a set of 256 contiguous class C networks.

An organization that decides to use IP addresses in the address space defined above can do so without any coordination with IANA or an Internet registry. The address space can thus be used privately by many independent organizations at the same time, with NAT operation enabled on their border routers.

5.3. Routing Across NAT

The router running NAT should not advertise the private networks to the backbone. Only the networks with global addresses may be known outside the stub. However, global information that NAT receives from the stub border router can be advertised in the stub the usual way.

Typically, the NAT stub router will have a static route configured to forward all external traffic to service provider router over WAN link, and the service provider router will have a static route configured to forward NAT packets (i.e., those whose destination IP address fall within the range of NAT managed global address list) to NAT router over WAN link.

5.4. Switch-over from Basic NAT to NAPT

In Basic NAT setup, when private network nodes outnumber global addresses available for mapping (say, a class B private network mapped to a class C global address block), external network access to some of the local nodes is abruptly cut off after the last global address from the address list is used up. This is very inconvenient and constraining. Such an incident can be safely avoided by optionally allowing the Basic NAT router to switch over to NAPT setup for the last global address in the address list. Doing this will ensure that hosts on private network will have continued, uninterrupted access to the external nodes and services for most applications. Note, however, it could be confusing if some of the applications that used to work with Basic NAT suddenly break due to the switch-over to NAPT.

6.0. NAT limitations

[NAT-TERM] covers the limitations of all flavors of NAT, broadly speaking. The following sub-sections identify limitations specific to traditional NAT.

6.1. Privacy and Security

Traditional NAT can be viewed as providing a privacy mechanism as sessions are uni-directional from private hosts and the actual addresses of the private hosts are not visible to external hosts.

The same characteristic that enhances privacy potentially makes debugging problems (including security violations) more difficult. If a host in private network is abusing the Internet in some way (such as trying to attack another machine or even sending large amounts of spam) it is more difficult to track the actual source of trouble because the IP address of the host is hidden in a NAT router.

6.2. ARP responses to NAT mapped global addresses on a LAN interface

NAT must be enabled only on border routers of a stub domain. The examples provided in the document to illustrate Basic NAT and NAPT have maintained a WAN link for connection to external router (i.e., service provider router) from NAT router. However, if the WAN link were to be replaced by a LAN connection and if part or all of the global address space used for NAT mapping belongs to the same IP subnet as the LAN segment, the NAT router would be expected to provide ARP support for the address range that belongs to the same subnet. Responding to ARP requests for the NAT mapped global addresses with its own MAC address is a must in such a situation with Basic NAT setup. If the NAT router did not respond to these requests, there is no other node in the network that has ownership to these addresses and hence will go unresponded.

This scenario is unlikely with NAPT setup except when the single address used in NAPT mapping is not the interface address of the NAT router (as in the case of a switch-over from Basic NAT to NAPT explained in 5.4 above, for example).

Using an address range from a directly connected subnet for NAT address mapping would obviate static route configuration on the service provider router.

It is the opinion of the authors that a LAN link to a service provider router is not very common. However, vendors may be interested to optionally support proxy ARP just in case.

6.3. Translation of outbound TCP/UDP fragmented packets in NAPT setup

Translation of outbound TCP/UDP fragments (i.e., those originating from private hosts) in NAPT setup are doomed to fail. The reason is as follows. Only the first fragment contains the TCP/UDP header that would be necessary to associate the packet to a session for translation purposes. Subsequent fragments do not contain TCP/UDP port information, but simply carry the same fragmentation identifier specified in the first fragment. Say, two private hosts originated fragmented TCP/UDP packets to the same destination host. And, they happened to use the same fragmentation identifier. When the target host receives the two unrelated datagrams, carrying same fragmentation id, and from the same assigned host address, it is unable to determine which of the two sessions the datagrams belong to. Consequently, both sessions will be corrupted.

7.0. Current Implementations

Many commercial implementations are available in the industry that adhere to the NAT description provided in this document. Linux public domain software contains NAT under the name of "IP masquerade". FreeBSD public domain software has NAPT implementation running as a daemon. Note however that Linux source is covered under the GNU license and FreeBSD software is covered under the UC Berkeley license.

Both Linux and FreeBSD software are free, so you can buy CD-ROMs for these for little more than the cost of distribution. They are also available on-line from a lot of FTP sites with the latest patches.

8.0. Security Considerations

The security considerations described in [NAT-TERM] for all variations of NATs are applicable to traditional NAT.

References

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

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

[RFC 1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

[RFC 1122] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[RFC 1123] Braden, R., "Requirements for Internet Hosts -- Application and Support", STD 3, RFC 1123, October 1989.

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

[FTP] Postel, J. and J. Reynolds, "FILE TRANSFER PROTOCOL (FTP)", STD 9, RFC 959, October 1985.

[TCP] Defense Advanced Research Projects Agency Information Processing Techniques Office, "TRANSMISSION CONTROL PROTOCOL (TCP) SPECIFICATION", STD 7, RFC 793, September 1981.

[ICMP] Postel, J., "INTERNET CONTROL MESSAGE (ICMP) SPECIFICATION", STD 5, RFC 792, September 1981.

[UDP] Postel, J., "User Datagram Protocol (UDP)", STD 6, RFC 768, August 1980.

[RFC 2101] Carpenter, B., Crowcroft, J. and Y. Rekhter, "IPv4 Address Behaviour Today", RFC 2101, February 1997.

Authors' Addresses

Pyda Srisuresh
Jasmine Networks, Inc.
3061 Zanker Road, Suite B
San Jose, CA 95134
U.S.A.

Phone: (408) 895-5032
EMail: srisuresh@yahoo.com

Kjeld Borch Egevang
Intel Denmark ApS

Phone: +45 44886556
Fax: +45 44886051
EMail: kjeld.egevang@intel.com
http: //www.freeyellow.com/members/kbe

Acknowledgement

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