

Architectural Implications of NAT

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

Abstract

In light of the growing interest in, and deployment of network address translation (NAT) RFC-1631, this paper will discuss some of the architectural implications and guidelines for implementations. It is assumed the reader is familiar with the address translation concepts presented in RFC-1631.

Table of Contents

1. Introduction.....	2
2. Terminology.....	4
3. Scope.....	6
4. End-to-End Model.....	7
5. Advantages of NATs.....	8
6. Problems with NATs.....	10
7. Illustrations.....	13
7.1 Single point of failure.....	13
7.2. ALG complexity.....	14
7.3. TCP state violations.....	15
7.4. Symmetric state management.....	16
7.5. Need for a globally unique FQDN when advertising public services.....	18
7.6. L2TP tunnels increase frequency of address collisions.....	19
7.7. Centralized data collection system report correlation.....	20
8. IPv6.....	21
9. Security Considerations.....	22
10. Deployment Guidelines.....	23
11. Summary.....	24
12. References.....	27

13. Acknowledgments..... 28
 14. Author's Address..... 28
 15. Full Copyright Statement..... 29

1. Introduction

Published in May 1994, written by K. Egevang and P. Francis, RFC-1631 [2] defined NAT as one means to ease the growth rate of IPv4 address use. But the authors were worried about the impact of this technology. Several places in the document they pointed out the need to experiment and see what applications may be adversely affected by NAT's header manipulations, even before there was any significant operational experience. This is further evidenced in a quote from the conclusions: 'NAT has several negative characteristics that make it inappropriate as a long term solution, and may make it inappropriate even as a short term solution.'

Now, six years later and in spite of the prediction, the use of NATs is becoming widespread in the Internet. Some people are proclaiming NAT as both the short and long term solution to some of the Internet's address availability issues and questioning the need to continue the development of IPv6. The claim is sometimes made that NAT 'just works' with no serious effects except on a few legacy applications. At the same time others see a myriad of difficulties caused by the increasing use of NAT.

The arguments pro & con frequently take on religious tones, with each side passionate about its position.

- Proponents bring enthusiasm and frequently cite the most popular applications of Mail & Web services as shining examples of NAT transparency. They will also point out that NAT is the feature that finally breaks the semantic overload of the IP address as both a locator and the global endpoint identifier (EID).
- An opposing view of NAT is that of a malicious technology, a weed which is destined to choke out continued Internet development. While recognizing there are perceived address shortages, the opponents of NAT view it as operationally inadequate at best, bordering on a sham as an Internet access solution. Reality lies somewhere in between these extreme viewpoints.

In any case it is clear NAT affects the transparency of end-to-end connectivity for transports relying on consistency of the IP header, and for protocols which carry that address information in places other than the IP header. Using a patchwork of consistently configured application specific gateways (ALG's), endpoints can work around some of the operational challenges of NAT. These operational challenges vary based on a number of factors including network and

application topologies and the specific applications in use. It can be relatively easy to deal with the simplest case, with traffic between two endpoints running over an intervening network with no parallel redundant NAT devices. But things can quickly get quite complicated when there are parallel redundant NAT devices, or where there are more distributed and multi-point applications like multi-party document sharing. The complexity of coordinating the updates necessary to work around NAT grows geometrically with the number of endpoints. In a large environment, this may require concerted effort to simultaneously update all endpoints of a given application or service.

The architectural intent of NAT is to divide the Internet into independent address administrations, (also see "address realms", RFC-2663 [3]) specifically facilitating casual use of private address assignments RFC-1918 [4]. As noted by Carpenter, et al RFC-2101 [5], once private use addresses were deployed in the network, addresses were guaranteed to be ambiguous. For example, when simple NATs are inserted into the network, the process of resolving names to or from addresses becomes dependent on where the question was asked. The result of this division is to enforce a client/server architecture (vs. peer/peer) where the servers need to exist in the public address realm.

A significant factor in the success of the Internet is the flexibility derived from a few basic tenets. Foremost is the End-to-End principle (discussed further below), which notes that certain functions can only be performed in the endpoints, thus they are in control of the communication, and the network should be a simple datagram service that moves bits between these points. Restated, the endpoint applications are often the only place capable of correctly managing the data stream. Removing this concern from the lower layer packet-forwarding devices streamlines the forwarding process, contributing to system-wide efficiency.

Another advantage is that the network does not maintain per connection state information. This allows fast rerouting around failures through alternate paths and to better scaling of the overall network. Lack of state also removes any requirement for the network nodes to notify each other as endpoint connections are formed or dropped. Furthermore, the endpoints are not, and need not be, aware of any network components other than the destination, first hop router(s), and an optional name resolution service. Packet integrity is preserved through the network, and transport checksums and any address-dependent security functions are valid end-to-end.

NAT devices (particularly the NAPT variety) undermine most of these, basic advantages of the end-to-end model, reducing overall flexibility, while often increasing operational complexity and impeding diagnostic capabilities. NAT variants such as RSIP [6] have recently been proposed to address some of the end-to-end concerns. While these proposals may be effective at providing the private node with a public address (if ports are available), they do not eliminate several issues like network state management, upper layer constraints like TCP_TIME_WAIT state, or well-known-port sharing. Their port multiplexing variants also have the same DNS limitations as NAPT, and each host requires significant stack modifications to enable the technology (see below).

It must be noted that firewalls also break the end-to-end model and raise several of the same issues that NAT devices do, while adding a few of their own. But one operational advantage with firewalls is that they are generally installed into networks with the explicit intent to interfere with traffic flow, so the issues are more likely to be understood or at least looked at if mysterious problems arise. The same issues with NAT devices can sometimes be overlooked since NAT devices are frequently presented as transparent to applications.

One thing that should be clearly stated up front is, that attempts to use a variant of NAT as a simple router replacement may create several significant issues that should be addressed before deployment. The goal of this document is to discuss these with the intent to raise awareness.

2. Terminology

Recognizing that many of these terms are defined in detail in RFC 2663 [3], the following are summaries as used in this document.

NAT - Network Address Translation in simple form is a method by which IP addresses are mapped from one address administration to another. The NAT function is unaware of the applications traversing it, as it only looks at the IP headers.

ALG - Application Layer Gateway: inserted between application peers to simulate a direct connection when some intervening protocol or device prevents direct access. It terminates the transport protocol, and may modify the data stream before forwarding.

NAT/ALG - combines ALG functions with simple NAT. Generally more useful than pure NAT, because it embeds components for specific applications that would not work through a pure NAT.

DNS/ALG - a special case of the NAT/ALG, where an ALG for the DNS service interacts with the NAT component to modify the contents of a DNS response.

Firewall - access control point that may be a special case of an ALG, or packet filter.

Proxy - A relay service designed into a protocol, rather than arbitrarily inserted. Unlike an ALG, the application on at least one end must be aware of the proxy.

Static NAT - provides stable one-to-one mapping between address spaces.

Dynamic NAT - provides dynamic mapping between address spaces normally used with a relatively large number of addresses on one side (private space) to a few addresses on the other (public space).

NAPT - Network Address Port Translation accomplishes translation by multiplexing transport level identifiers of multiple addresses from one side, simultaneously into the transport identifiers of a single address on the other. See 4.1.2 of RFC-2663. This permits multiple endpoints to share and appear as a single IP address.

RSIP - Realm Specific IP allows endpoints to acquire and use the public address and port number at the source. It includes mechanisms for the private node to request multiple resources at once. RSIP clients must be aware of the address administration boundaries, which specific administrative area its peer resides in for each application, and the topology for reaching the peer. To complete a connection, the private node client requests one or more addresses and/or ports from the appropriate RSIP server, then initiates a connection via that RSIP server using the acquired public resources. Hosts must be updated with specific RSIP software to support the tunneling functions.

VPN - For purposes of this document, Virtual Private Networks technically treat an IP infrastructure as a multiplexing substrate, allowing the endpoints to build virtual transit pathways, over which they run another instance of IP. Frequently the 2nd instance of IP uses a different set of IP addresses.

AH - IP Authentication Header, RFC-2401 [7], which provides data integrity, data origin authentication, and an optional anti-replay service.

ESP - Encapsulating Security Payload protocol, RFC 2401, may provide data confidentiality (encryption), and limited traffic flow confidentiality. It also may provide data integrity, data origin authentication, and an anti-replay service.

Address administration - coordinator of an address pool assigned to a collection of routers and end systems.

Addressing realm - a collection of routers and end systems exchanging locally unique location knowledge. (Further defined in RFC-2663 NAT Terminology.) NAT is used as a means to distribute address allocation authority and provide a mechanism to map addresses from one address administration into those of another administration.

3. Scope

In discussing the architectural impact of NATs on the Internet, the first task is defining the scope of the Internet. The most basic definition is; a concatenation of networks built using IETF defined technologies. This simple description does not distinguish between the public network known as the Internet, and the private networks built using the same technologies (including those connected via NAT). Rekhter, et al in RFC-1918 defined hosts as public when they need network layer access outside the enterprise, using a globally unambiguous address. Those that need limited or no access are defined as private. Another way to view this is in terms of the transparency of the connection between any given node and the rest of the Internet.

The ultimate resolution of public or private is found in the intent of the network in question. Generally, networks that do not intend to be part of the greater Internet will use some screening technology to insert a barrier. Historically barrier devices between the public and private networks were known as Firewalls or Application Gateways, and were managed to allow approved traffic while blocking everything else. Increasingly, part of the screening technology is a NAT, which manages the network locator between the public and private-use address spaces, and then, using ALGs adds support for protocols that are incompatible with NAT. (Use of NAT within a private network is possible, and is only addressed here in the context that some component of the private network is used as a common transit service between the NAT attached stubs.)

RFC-1631 limited the scope of NAT discussions to stub appendages of a public Internet, that is, networks with a single connection to the rest of the Internet. The use of NAT in situations in which a network has multiple connections to the rest of the Internet is significantly more complex than when there is only a single

connection since the NATs have to be coordinated to ensure that they have a consistent understanding of address mapping for each individual device.

4. End-to-End Model

The concept of the End-to-End model is reviewed by Carpenter in Internet Transparency [8]. One of the key points is "state should be maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks"; this is termed "fate-sharing". The goal behind fate-sharing is to ensure robustness. As networks grow in size, likelihood of component failures affecting a connection becomes increasingly frequent. If failures lead to loss of communication, because key state is lost, then the network becomes increasingly brittle, and its utility degrades. However, if an endpoint itself fails, then there is no hope of subsequent communication anyway. Therefore the End-to-End model argues that as much as possible, only the endpoints should hold critical state.

For NATs, this aspect of the End-to-End model translates into the NAT becoming a critical infrastructure element: if it fails, all communication through it fails, and, unless great care is taken to assure consistent, stable storage of its state, even when it recovers the communication that was passing through it will still fail (because the NAT no longer translates it using the same mappings). Note that this latter type of failure is more severe than the failure of a router; when a router recovers, any communication that it had been forwarding previous can continue to be successfully forwarded through it.

There are other important facets to the End-to-End model:

- when state is held in the interior of the network, then traffic dependent on that state cannot be routed around failures unless somehow the state is replicated to the fail-over points, which can be very difficult to do in a consistent yet efficient and timely fashion.
- a key principle for scaling networks to large size is to push state-holding out to the edges of the network. If state is held by elements in the core of the network, then as the network grows the amount of state the elements must hold likewise grows. The capacities of the elements can become severe chokepoints and the number of connections affected by a failure also grows.
- if security state must be held inside the network (see the discussion below), then the possible trust models the network can support become restricted.

A network for which endpoints need not trust network service providers has a great deal more security flexibility than one which does. (Picture, for example, a business traveler connecting from their hotel room back to their home office: should they have to trust the hotel's networking staff with their security keys?, or the staff of the ISP that supplies the hotel with its networking service? How about when the traveler connects over a wireless connection at an airport?)

Related to this, RFC-2101 notes:

Since IP Security authentication headers assume that the addresses in the network header are preserved end-to-end, it is not clear how one could support IP Security-based authentication between a pair of hosts communicating through either an ALG or a NAT.

In addition, there are distributed applications that assume that IP addresses are globally scoped, globally routable, and all hosts and applications have the same view of those addresses. Indeed, a standard technique for such applications to manage their additional control and data connections is for one host to send to another the address and port that the second host should connect to. NATs break these applications. Similarly, there are other applications that assume that all upper layer ports from a given IP address map to the same endpoint, and port multiplexing technologies like NATPT and RSIP break these. For example, a web server may desire to associate a connection to port 80 with one to port 443, but due to the possible presence of a NATPT, the same IP address no longer ensures the same host.

Limiting such applications is not a minor issue: much of the success of the Internet today is due to the ease with which new applications can run on endpoints without first requiring upgrades to infrastructure elements. If new applications must have the NATs upgraded in order to achieve widespread deployment, then rapid deployment is hindered, and the pace of innovation slowed.

5. Advantages of NATs

A quick look at the popularity of NAT as a technology shows that it tackles several real world problems when used at the border of a stub domain.

- By masking the address changes that take place, from either dial-access or provider changes, minimizes impact on the local network by avoiding renumbering.
- Globally routable addresses can be reused for intermittent access customers. This pushes the demand for addresses towards the number of active nodes rather than the total number of nodes.

- There is a potential that ISP provided and managed NATs would lower support burden since there could be a consistent, simple device with a known configuration at the customer end of an access interface.
- Breaking the Internet into a collection of address authorities limits the need for continual justification of allocations allows network managers to avoid the use of more advanced routing techniques such as variable length subnets.
- Changes in the hosts may not be necessary for applications that don't rely on the integrity of the packet header, or carry IP addresses in the payload.
- Like packet filtering Firewalls, NAT, & RSIP block inbound connections to all ports until they are administratively mapped.

Taken together these explain some of the strong motivations for moving quickly with NAT deployment. Traditional NAT [2] provides a relatively simple function that is easily understood.

Removing hosts that are not currently active lowers address demands on the public Internet. In cases where providers would otherwise end up with address allocations that could not be aggregated, this improves the load on the routing system as well as lengthens the lifetime of the IPv4 address space. While reclaiming idle addresses is a natural byproduct of the existing dynamic allocation, dial-access devices, in the dedicated connection case this service could be provided through a NAT. In the case of a NAT, the aggregation potential is even greater as multiple end systems share a single public address.

By reducing the potential customer connection options and minimizing the support matrix, it is possible that ISP provided NATs would lower support costs.

Part of the motivation for NAT is to avoid the high cost of renumbering inherent in the current IPv4 Internet. Guidelines for the assignment of IPv4 addresses RFC-2050 [9] mean that ISP customers are currently required to renumber their networks if they want to switch to a new ISP. Using a NAT (or a firewall with NAT functions) means that only the Internet facing IP addresses must be changed and internal network nodes do not need to be reconfigured. Localizing address administration to the NAT minimizes renumbering costs, and simultaneously provides for a much larger local pool of addresses than is available under current allocation guidelines. (The registry guidelines are intended to prolong the lifetime of the IPv4 address space and manage routing table growth, until IPv6 is ready or new routing technology reduces the pressure on the routing table. This is accomplished by managing allocations to match actual demand and to enforce hierarchical addressing. An unfortunate byproduct of the

current guidelines is that they may end up hampering growth in areas where it is difficult to sort out real need from potential hoarding.) NAT is effective at masking provider switching or other requirements to change addresses, thus mitigates some of the growth issues.

NAT deployments have been raising the awareness of protocol designers who are interested in ensuring that their protocols work end-to-end. Breaking the semantic overload of the IP address will force applications to find a more appropriate mechanism for endpoint identification and discourage carrying the locator in the data stream. Since this will not work for legacy applications, RFC-1631 discusses how to look into the packet and make NAT transparent to the application (i.e.: create an application gateway). This may not be possible for all applications (such as IP based authentication in SNMP), and even with application gateways in the path it may be necessary to modify each end host to be aware when there are intermediaries modifying the data.

Another popular practice is hiding a collection of hosts that provide a combined service behind a single IP address (i.e.: web host load sharing). In many implementations this is architecturally a NAT, since the addresses are mapped to the real destination on the fly. When packet header integrity is not an issue, this type of virtual host requires no modifications to the remote applications since the end client is unaware of the mapping activity. While the virtual host has the CPU performance characteristics of the total set of machines, the processing and I/O capabilities of the NAT/ALG device bound the overall performance as it funnels the packets back and forth.

6. Problems with NATs

- NATs break the flexible end-to-end model of the Internet.
- NATs create a single point where fates are shared, in the device maintaining connection state and dynamic mapping information.
- NATs complicate the use of multi-homing by a site in order to increase the reliability of their Internet connectivity. (While single routers are a point of fate sharing, the lack of state in a router makes creating redundancy trivial. Indeed, this is one of the reasons why the Internet protocol suite developed using a connectionless datagram service as its network layer.)
- NATs inhibit implementation of security at the IP level.
- NATs enable casual use of private addresses. These uncoordinated addresses are subject to collisions when companies using these addresses merge or want to directly interconnect using VPNs.
- NATs facilitate concatenating existing private name spaces with the public DNS.

- Port versions (NAPT and RSIP) increase operational complexity when publicly published services reside on the private side.
- NATs complicated or may even invalidate the authentication mechanism of SNMPv3.
- Products may embed a NAT function without identifying it as such.

By design, NATs impose limitations on flexibility. As such, extended thought about the introduced complications is called for. This is especially true for products where the NAT function is a hidden service, such as load balancing routers that re-write the IP address to other public addresses. Since the addresses may be all in publicly administered space these are rarely recognized as NATs, but they break the integrity of the end-to-end model just the same.

NATs place constraints on the deployment of applications that carry IP addresses (or address derivatives) in the data stream, and they operate on the assumption that each session is independent. However, there are applications such as FTP and H.323 that use one or more control sessions to set the characteristics of the follow-on sessions in their control session payload. Other examples include SNMP MIBs for configuration, and COPS policy messages. Applications or protocols like these assume end-to-end integrity of addresses and will fail when traversing a NAT. (TCP was specifically designed to take advantage of, and reuse, the IP address in combination with its ports for use as a transport address.) To fix how NATs break such applications, an Application Level Gateway needs to exist within or alongside each NAT. An additional gateway service is necessary for each application that may imbed an address in the data stream. The NAT may also need to assemble fragmented datagrams to enable translation of the application stream, and then adjust TCP sequence numbers, prior to forwarding.

As noted earlier, NATs break the basic tenet of the Internet that the endpoints are in control of the communication. The original design put state control in the endpoints so there would be no other inherent points of failure. Moving the state from the endpoints to specific nodes in the network reduces flexibility, while it increases the impact of a single point failure. See further discussion in Illustration 1 below.

In addition, NATs are not transparent to all applications, and managing simultaneous updates to a large array of ALGs may exceed the cost of acquiring additional globally routable addresses. See further discussion in Illustration 2 below.

While RSIP addresses the transparency and ALG issues, for the specific case of an individual private host needing public access, there is still a node with state required to maintain the connection.

Dynamic NAT and RSIP will eventually violate higher layer assumptions about address/port number reuse as defined in RFC-793 [10] RFC-1323 [11]. The TCP state, TCP_TIME_WAIT, is specifically designed to prevent replay of packets between the 4-tuple of IP and port for a given IP address pair. Since the TCP state machine of a node is unaware of any previous use of RSIP, its attempt to connect to the same remote service that its neighbor just released (which is still in TCP_TIME_WAIT) may fail, or with a larger sequence number may open the prior connection directly from TCP_TIME_WAIT state, at the loss of the protection afforded by the TCP_TIME_WAIT state (further discussion in 2.6 of RFC-2663 [3]).

For address translators (which do not translate ports) to comply with the TCP_TIME_WAIT requirements, they must refrain from assigning the same address to a different host until a period of 2*MSL has elapsed since the last use of the address, where MSL is the Maximum Segment Lifetime defined in RFC-793 as two minutes. For address-and-port translators to comply with this requirement, they similarly must refrain from assigning the same host/port pair until 2*MSL has elapsed since the end of its first use. While these requirements are simple to state, they can place a great deal of pressure on the NAT, because they temporarily reduce the pool of available addresses and ports. Consequently, it will be tempting or NAT implementers to ignore or shorten the TCP_TIME_WAIT requirements, at the cost of some of TCP's strong reliability. Note that in the case where the strong reliability is in fact compromised by the appearance of an old packet, the failure can manifest itself as the receiver accepting incorrect data. See further discussion in Illustration 3 below.

It is sometimes argued that NATs simply function to facilitate "routing realms", where each domain is responsible for finding addresses within its boundaries. Such a viewpoint clouds the limitations created by NAT with the better-understood need for routing management. Compartmentalization of routing information is correctly a function of routing protocols and their scope of application. NAT is simply a means to distribute address allocation authority and provide a mechanism to map addresses from one address realm into those of another realm.

In particular, it is sometimes erroneously believed that NATs serve to provide routing isolation. In fact, if someone were to define an OSPF ALG it would actually be possible to route across a NAT boundary. Rather than NAT providing the boundary, it is the experienced operators who know how to limit network topology that serve to avoid leaking addresses across a NAT. This is an operational necessity given the potential for leaked addresses to introduce inconsistencies into the public infrastructure.

One of the greatest concerns from the explosion of NATs is the impact on the fledgling efforts at deploying network layer end-to-end IP security. One fundamental issue for IPsec is that with both AH and ESP, the authentication check covers the TCP/UDP checksum (which in turn covers the IP address). When a NAT changes the IP address, the checksum calculation will fail, and therefore authentication is guaranteed to fail. Attempting to use the NAT as a security boundary fails when requirement is end-to-end network layer encryption, since only the endpoints have access to the keys. See further discussion in Illustration 4 below.

Finally, while the port multiplexing variants of NAT (popular because they allow Internet access through a single address) work modestly well for connecting private hosts to public services, they create management problems for applications connecting from public toward private. The concept of a well-known port is undermined because only one private side system can be mapped through the single public-side port number. This will affect home networks, when applications like multi-player Internet games can only be played on one system at a time. It will also affect small businesses when only one system at a time can be operated on the standard port to provide web services. These may sound like only medium-grade restrictions for the present, but as a basic property of the Internet, not to change years into the future, it is highly undesirable. The issue is that the public toward private usage requires administrative mapping for each target prior to connection. If the ISP chooses to provide a standardized version of these to lower configuration options, they may find the support costs of managing the ALGs will exceed the cost of additional address space. See further discussion in Illustration 6 below.

7. Illustrations

7.1 Single point of failure

A characteristic of stateful devices like NATs is the creation of a single point of failure. Attempts to avoid this by establishing redundant NATs, creates a new set of problems related to timely communication of the state, and routing related failures. This encompasses several issues such as update frequency, performance impact of frequent updates, reliability of the state update transaction, a-priori knowledge of all nodes needing this state information, and notification to end nodes of alternatives. (This notification could be accomplished with a routing protocol, which might require modifications to the hosts so they will listen.)

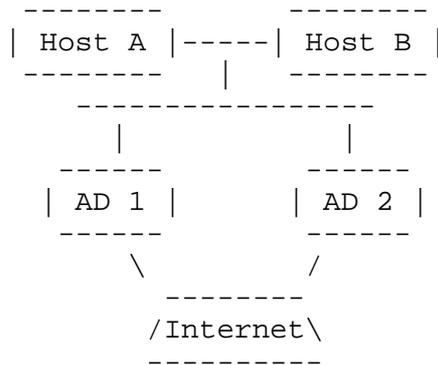


Illustration 1

In the traditional case where Access Device (AD) 1 & 2 are routers, the single point of failure is the end Host, and the only effort needed to maintain the connections through a router or link failure is a simple routing update from the surviving router. In the case where the ADs are a NAT variant there will be connection state maintained in the active path that would need to be shared with alternative NATs. When the Hosts have open connections through either NAT, and it fails, the application connections will drop unless the state had been previously moved to the surviving NAT. The hosts will still need to acquire a routing redirect. In the case of RSIP, the public side address pool would also need to be shared between the ADs to allow movement. This sharing creates another real-time operational complexity to prevent conflicting assignments at connection setup. NAT as a technology creates a point fate sharing outside the endpoints, in direct contradiction to the original Internet design goals.

7.2. ALG complexity

In the following example of a proposed corporate network, each NAT/ALG was to be one or more devices at each physical location, and there were to be multiple physical locations per diagrammed connection. The logistics of simply updating software on this scale is cumbersome, even when all the devices are the same manufacturer and model. While this would also be true with routers, it would be unnecessary for all devices to run a consistent version for an application to work across an arbitrary path.

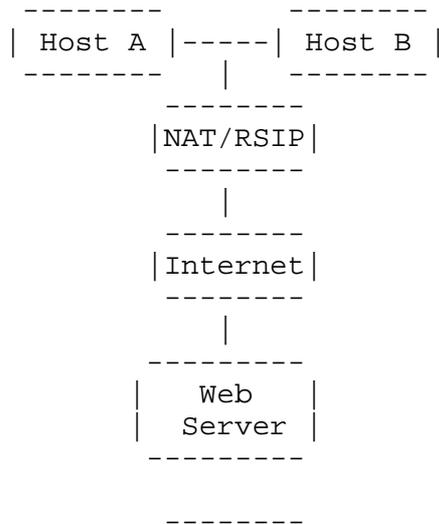


Illustration 3

Host A completes its transaction and closes the web service on TCP port 80, and the RSIP releases the public side address used for Host A. Host B attempts to open a connection to the same web service, and the NAT assigns then next free public side address which is the same one A just released. The source port selection rules on Host B happen to lead it to the same choice that A used. The connect request from Host B is rejected because the web server, conforming to the TCP specifications, has that 4-tuple in TIME WAIT for 4 minutes. By the time a call from Host B gets through to the helpdesk complaining about no access, the requested retry will work, so the issue is marked as resolved, when it in fact is an on-going, but intermittent, problem.

7.4. Symmetric state management

Operational management of networks incorporating stateful packet modifying device is considerably easier if inbound and outbound packets traverse the same path. (Otherwise it's a headache to keep state for the two directions synchronized.) While easy to say, even with careful planning it can be difficult to manage using a connectionless protocol like IP. The problem of creating redundant connections is ensuring that routes advertised to the private side reach the end nodes and map to the same device as the public side route advertisements. This state needs to persist throughout the lifetime of sessions traversing the NAT, in spite of frequent or simultaneous internal and external topology churn. Consider the following case where the -X- links are broken, or flapping.

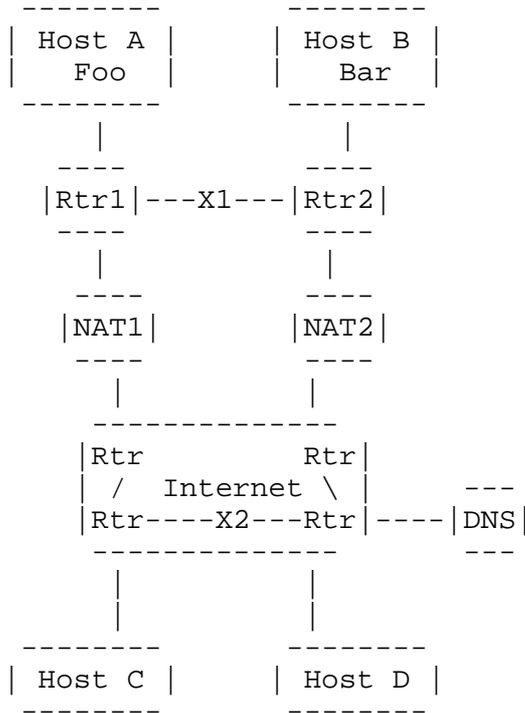


 Illustration 4

To preserve a consistent view of routing, the best path to the Internet for Routers 1 & 2 is via NAT1, while the Internet is told the path to the address pool managed by the NATs is best found through NAT1. When the path X1 breaks, Router 2 would attempt to switch to NAT2, but the external return path would still be through NAT1. This is because the NAT1 device is advertising availability of a pool of addresses. Directly connected routers in this same situation would advertise the specific routes that existed after the loss. In this case, redundancy was useless.

Consider the case that the path between Router 1 & 2 is up, and some remote link in the network X2 is down. It is also assumed that DNS returns addresses for both NATs when queried for Hosts A or B. When Host D tries to contact Host B, the request goes through NAT2, but due to the internal routing, the reply is through NAT1. Since the state information for this connection is in NAT2, NAT1 will provide a new mapping. Even if the remote path is restored, the connection will not be made because the requests are to the public IP of NAT2, while the replies are from the public IP of NAT1.

In a third case, both Host A & B want to contact Host D, when the remote link X2 in the Internet breaks. As long as the path X1 is down, Host B is able to connect, but Host A is cut off. Without a thorough understanding of the remote topology (unlikely since Internet providers tend to consider that sensitive proprietary information), the administrator of Hosts A & B would have no clue why one worked and the other didn't. As far as he can tell the redundant paths through the NATs are up but only one connection works. Again, this is due to lack of visibility to the topology that is inherent when a stateful device is advertising availability to a pool rather than the actual connected networks.

In any network topology, individual router or link failures may present problems with insufficient redundancy, but the state maintenance requirements of NAT present an additional burden that is not as easily understood or resolved.

7.5. Need for a globally unique FQDN when advertising public services

The primary feature of NATs is the 'simple' ability to connect private networks to the public Internet. When the private network exists prior to installing the NAT, it is unlikely and unnecessary that its name resolver would use a registered domain. As noted in RFC 1123 [12] DNS queries may be resolved via local multicast. Connecting the NAT device, and reconfiguring it's resolver to proxy for all external requests allows access to the public network by hosts on the private network. Configuring the public DNS for the set of private hosts that need inbound connections would require a registered domain (either private, or from the connecting ISP) and a unique name. At this point the partitioned name space is concatenated and hosts would have different names based on inside vs. outside queries.

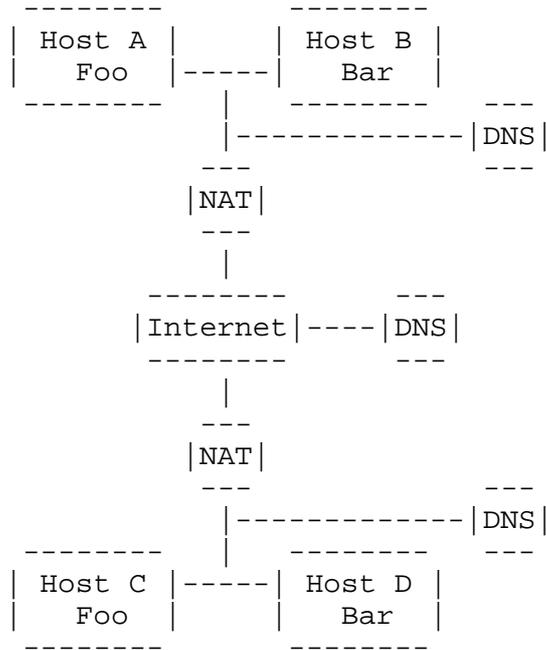


Illustration 5

Everything in this simple example will work until an application embeds a name. For example, a Web service running on Host D might present embedded URL's of the form `http://D/bar.html`, which would work from Host C, but would thoroughly confuse Host A. If the embedded name resolved to the public address, Host A would be happy, but Host C would be looking for some remote machine. Using the public FQDN resolution to establishing a connection from Host C to D, the NAT would have to look at the destination rather than simply forwarding the packet out to the router. (Normal operating mode for a NAT is translate & forward out the other interface, while routers do not send packets back on the same interface they came from.) The NAT did not create the name space fragmentation, but it facilitates attempts to merge networks with independent name administrations.

7.6. L2TP tunnels increase frequency of address collisions

The recent mass growth of the Internet has been driven by support of low cost publication via the web. The next big push appears to be support of Virtual Private Networks (VPNs) frequently accomplished using L2TP. Technically VPN tunnels treat an IP infrastructure as a multiplexing substrate allowing the endpoints to build what appear to be clear pathways from end-to-end. These tunnels redefine network visibility and increase the likelihood of address collision when

traversing multiple NATs. Address management in the private space behind NATs will become a significant burden, as there is no central body capable of, or willing to do it. The lower burden for the ISP is actually a transfer of burden to the local level, because administration of addresses and names becomes both distributed and more complicated.

As noted in RFC-1918, the merging of private address spaces can cause an overlap in address use, creating a problem. L2TP tunnels will increase the likelihood and frequency of that merging through the simplicity of their establishment. There are several configurations of address overlap which will cause failure, but in the simple example shown below the private use address of Host B matches the private use address of the VPN pool used by Host A for inbound connections. When Host B tries to establish the VPN interface, Host A will assign it an address from its pool for inbound connections, and identify the gateway for Host B to use. In the example, Host B will not be able to distinguish the remote VPN gateway address of Host A from its own private address on the physical interface, thus the connection will fail. Since private use addresses are by definition not publicly coordinated, as the complexity of the VPN mesh increases so does the likelihood that there will be a collision that cannot be resolved.

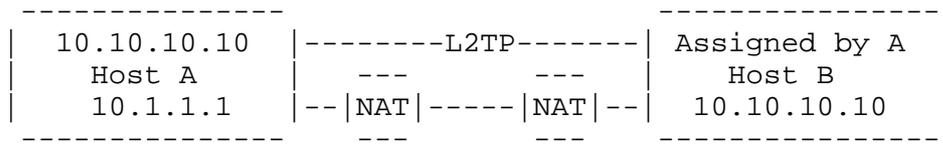


 Illustration 6

7.7. Centralized data collection system report correlation

It has been reported that NAT introduces additional challenges when intrusion detection systems attempt to correlate reports between sensors inside and outside the NAT. While the details of individual systems are beyond the scope of this document, it is clear that a centralized system with monitoring agents on both sides of the NAT would also need access to the current NAT mappings to get this right. It would also be critical that the resulting data be indexed properly if there were agents behind multiple NATs using the same address range for the private side.

This also applies to management data collected via SNMP. Any time the data stream carries an IP address; the central collector or ALG will need to manipulate the data based on the current mappings in the

NAT.

8. IPv6

It has been argued that IPv6 is no longer necessary because NATs relieve the address space constraints and allow the Internet to continue growing. The reality is they point out the need for IPv6 more clearly than ever. People are trying to connect multiple machines through a single access line to their ISP and have been willing to give up some functionality to get that at minimum cost.

Frequently the reason for cost increases is the perceived scarcity (therefore increased value) of IPv4 addresses, which would be eliminated through deployment of IPv6. This crisis mentality is creating a market for a solution to a problem already solved with greater flexibility by IPv6.

If NAT had never been defined, the motivation to resolve the dwindling IPv4 address space would be a much greater. Given that NATs are enabling untold new hosts to attach to the Internet daily, it is difficult to ascertain the actual impact to the lifetime of IPv4, but NAT has certainly extended it. It is also difficult to determine the extent of delay NAT is causing for IPv6, both by relieving the pressure, and by redirecting the intellectual cycles away from the longer-term solution.

But at the same time NAT functionality may be a critical facilitator in the deployment of IPv6. There are already 100 million or more computers running IPv4 on data networks. Some of these networks are connected to and thus part of the Internet and some are on private isolated networks. It is inconceivable that we could have a "flag day" and convert all of the existing IPv4 nodes to IPv6 at the same time. There will be a very long period of coexistence while both IPv4 and IPv6 are being used in the Internet and in private networks. The original IPv6 transition plan relied heavily on having new IPv6 nodes also be able to run IPv4 - a "dual stack" approach. When the dual stack node looks up another node in the DNS it will get back a IPv4 or an IPv6 address in response. If the response is an IPv4 address then the node uses IPv4 to contact the other node. And if the response is an IPv6 address then IPv6 can be used to make the contact. Turning the NAT into a 6to4 [13]router enables widespread deployment of IPv6 while providing an IPv4 path if IPv6 is unavailable. While this maintains the current set of issues for IPv4 connections, it reestablishes the end-to-end principle for IPv6 connections.

An alternative methodology would be to translate the packets between IPv6 and IPv4 at the borders between IPv4 supporting networks and IPv6 supporting networks. The need for this functionality was recognized in [RFC 1752], the document that recommended to the IETF that IPv6 be developed and recommended that a set of working groups be established to work on a number of specific problems. Header translation (i.e., NAT) was one of those problems.

Of course, NATs in an IPv6 to IPv4 translation environment encounter all of the same problems that NATs encounter in a pure IPv4 and the environment and cautions in this document apply to both situations.

9. Security Considerations

NAT (particularly NATPT) actually has the potential to lower overall security because it creates the illusion of a security barrier, but does so without the managed intent of a firewall. Appropriate security mechanisms are implemented in the end host, without reliance on assumptions about routing hacks, firewall filters, or missing NAT translations, which may change over time to enable a service to a neighboring host. In general, defined security barriers assume that any threats are external, leading to practices that make internal breaches much easier.

IPsec RFC-2401 [7] defines a set of mechanisms to support packet-level authentication and encryption for use in IP networks. While this may be less efficient than application-level security but in the words of RFC-1752 [14] "support for basic packet-level authentication will provide for the adoption of a much needed, widespread, security infrastructure throughout the Internet."

NATs break IPsec's authentication and encryption technologies because these technologies depend on an end-to-end consistency of the IP addresses in the IP headers, and therefore may stall further deployment of enhanced security across the Internet. NATs raise a number of specific issues with IPsec. For example;

- Use of AH is not possible via NAT as the hash protects the IP address in the header.
- Authenticated certificates may contain the IP address as part of the subject name for authentication purposes.
- Encrypted Quick Mode structures may contain IP addresses and ports for policy verifications.
- The Revised Mode of public key encryption includes the peer identity in the encrypted payload.

It may be possible to engineer and work around NATs for IPsec on a case-by-case basis, but at the cost of restricting the trust model, as discussed in section 4 above. With all of the restrictions placed on deployment flexibility, NATs present a significant obstacle to security integration being deployed in the Internet today.

As noted in the RFC-2694 [15], the DNS/ALG cannot support secure DNS name servers in the private domain. Zone transfers between DNSsec servers will be rejected when necessary modifications are attempted. It is also the case that DNS/ALG will break any modified, signed responses. This would be the case for all public side queries of private nodes, when the DNS server is on the private side. It would also be true for any private side queries for private nodes, when the DNS server is on the public side. Digitally signed records could be modified by the DNS/ALG if it had access to the source authentication key. DNSsec has been specifically designed to avoid distribution of this key, to maintain source authenticity. So NATs that use DNS/ALG to repair the namespace resolutions will either; break the security when modifying the record, or will require access to all source keys to requested resolutions.

Security mechanisms that do not protect or rely on IP addresses as identifiers, such as TLS [16], SSL [17], or SSH [18] may operate in environments containing NATs. For applications that can establish and make use of this type of transport connection, NATs do not create any additional complications. These technologies may not provide sufficient protection for all applications as the header is exposed, allowing subversive acts like TCP resets. RFC-2385 [19] discusses the issues in more detail.

Arguments that NATs may operate in a secure mode preclude true End-to-End security, as the NAT becomes the security endpoint. Operationally the NAT must be managed as part of the security domain, and in this mode the packets on the unsecured side of the NAT are fully exposed.

10. Deployment Guidelines

Given that NAT devices are being deployed at a fairly rapid pace, some guidelines are in order. Most of these cautionary in nature and are designed to make sure that the reader fully understands the implications of the use of NATs in their environment.

- Determine the mechanism for name resolution, and ensure the appropriate answer is given for each address administration. Embedding the DNS server, or a DNS ALG in the NAT device will likely be more manageable than trying to synchronize independent DNS systems across administrations.

- Is the NAT configured for static one to one mappings, or will it dynamically manage them? If dynamic, make sure the TTL of the DNS responses is set to 0, and that the clients pay attention to the don't cache notification.
- Will there be a single NAT device, or parallel with multiple paths? If single, consider the impact of a device failure. If multiple, consider how routing on both sides will insure the packets flow through the same box over the connection lifetime of the applications.
- Examine the applications that will need to traverse the NAT and verify their immunity to address changes. If necessary provide an appropriate ALG or establish a VPN to isolate the application from the NAT.
- Determine need for public toward private connections, variability of destinations on the private side, and potential for simultaneous use of public side port numbers. NATs increase administration if these apply.
- Determine if the applications traversing the NAT or RSIP expect all ports from the public IP address to be the same endpoint. Administrative controls to prevent simultaneous access from multiple private hosts will be required if this is the case.
- If there are encrypted payloads, the contents cannot be modified unless the NAT is a security endpoint, acting as a gateway between security realms. This precludes end-to-end confidentiality, as the path between the NAT and endpoint is exposed.
- Determine the path for name resolutions. If hosts on the private side of a NAT or RSIP server need visibility to each other, a private side DNS server may be required.
- If the environment uses secure DNS records, the DNS/ALG will require access to the source authentication keys for all records to be translated.
- When using VPNs over NATs, identify a clearinghouse for the private side addresses to avoid collisions.
- Assure that applications used both internally and externally avoid embedding names, or use globally unique ones.
- When using RSIP, recognize the scope is limited to individual private network connecting to the public Internet. If other NATs are in the path (including web-server load-balancing devices), the advantage of RSIP (end-to-end address/port pair use) is lost.
- For RSIP, determine the probability of TCP_Time_Wait collisions when subsequent private side hosts attempt to contact a recently disconnected public side service.

11. Summary

Over the 6-year period since RFC-1631, the experience base has grown, further exposing concerns raised by the original authors. NAT breaks a fundamental assumption of the Internet design; the endpoints are in control. Another design principle, 'keep-it-simple' is being overlooked as more features are added to the network to work around the complications created by NATs. In the end, overall flexibility and manageability are lowered, and support costs go up to deal with the problems introduced.

Evangelists, for and against the technology, present their cases as righteous while downplaying any rebuttals.

- NATs are a 'fact of life', and will proliferate as an enhancement that sustains the existing IPv4 infrastructure.
- NATs are a 'necessary evil' and create an administrative burden that is not easily resolved. More significantly, they inhibit the roll out of IPsec, which will in turn slow growth of applications that require a secure infrastructure.

In either case, NATs require strong applicability statements, clearly declaring what works and what does not.

An overview of the pluses and minuses:

NAT advantages	NAT disadvantages
-----	-----
Masks global address changes	Breaks end-to-end model
Eases renumbering when providers change	Facilitates concatenation of multiple name spaces
	Breaks IPsec
	Stateful points of failure
Address administrations avoid justifications to registries	Requires source specific DNS reply or DNS/ALG
	DNS/ALG breaks DNSsec replies
Lowers address utilization	Enables end-to-end address conflicts
	Increases local support burden and complexity
Lowers ISP support burden	Unique development for each app
Transparent to end systems in some cases	
Load sharing as virtual host	Performance limitations with scale
Delays need for IPv4 replacement	May complicate integration of IPv6

There have been many discussions lately about the value of continuing with IPv6 development when the market place is widely deploying IPv4 NATs. A shortsighted view would miss the point that both have a role, because NATs address some real-world issues today, while IPv6 is targeted at solving fundamental problems, as well as moving forward. It should be recognized that there will be a long co-existence as applications and services develop for IPv6, while the lifetime of the existing IPv4 systems will likely be measured in decades. NATs are a diversion from forward motion, but they do enable wider participation at the present state. They also break a class of applications, which creates the need for complex work-around scenarios.

Efforts to enhance general security in the Internet include IPsec and DNSsec. These technologies provide a variety of services to both authenticate and protect information during transit. By breaking these technologies, NAT and the DNS/ALG work-around, hinder deployment of enhanced security throughout the Internet.

There have also been many questions about the probability of VPNs being established that might raise some of the listed concerns. While it is hard to predict the future, one way to avoid ALGs for each application is to establish a L2TP over the NATs. This restricts the NAT visibility to the headers of the tunnel packets, and removes its effects from all applications. While this solves the ALG issues, it raises the likelihood that there will be address collisions as arbitrary connections are established between uncoordinated address spaces. It also creates a side concern about how an application establishes the necessary tunnel.

The original IP architecture is powerful because it provides a general mechanism on which other things (yet unimagined) may be built. While it is possible to build a house of cards, time and experience have lead to building standards with more structural integrity. IPv6 is the long-term solution that retains end-to-end transparency as a principle. NAT is a technological diversion to sustain the lifetime of IPv4.

12. References

- 1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.
- 2 Egevang, K. and P. Francis, "The IP Network Address Translator", RFC 1631, May 1994.
- 3 Srisuresh, P. and M. Holdrege, "NAT Terminology and Considerations", RFC 2663, August 1999.
- 4 Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- 5 Carpenter, B., Crowcroft, J. and Y. Rekhter, "IPv4 Address Behavior Today", RFC 2101, February 1997.
- 6 M. Borella, D. Grabelsky, J., K. Tuniguchi, "Realm Specific IP: Protocol Specification", Work in Progress, March 2000.
- 7 Kent, S. and R. Atkinson, "Security Architecture for IP", RFC 2401, November 1998.
- 8 Carpenter, B., "Internet Transparency", RFC 2775, February 2000.
- 9 Hubbard, K., Koster, M., Conrad, D., Karrenberg, D. and J. Postel, "Internet Registry IP Allocation Guidelines", BCP 12, RFC 2050, November 1996.
- 10 Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- 11 Jacobson, V., Braden, R. and L. Zhang, "TCP Extension for High-Speed Paths", RFC 1185, October 1990.
- 12 Braden, R., "Requirements for Internet Hosts", STD 3, RFC 1123, October 1989.
- 13 Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds without Explicit Tunnels", Work in Progress.
- 14 Bradner, S. and A. Mankin, "Recommendation for IPng", RFC 1752, January 1995.
- 15 Srisuresh, P., Tsirtsis, G., Akkiraju, P. and A. Heffernan, "DNS extensions to NAT", RFC 2694, September 1999.

- 16 Dierks, T. and C. Allen, "The TLS Protocol", RFC 2246, January 1999.
- 17 <http://home.netscape.com/eng/ssl3/ssl-toc.html>, March 1996.
- 18 T. Ylonen, et al., "SSH Protocol Architecture", Work in Progress, August 1998.
- 19 Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

13. Acknowledgments

Valuable contributions to this document came from the IAB, Vern Paxson (lbl), Scott Bradner (harvard), Keith Moore (utk), Thomas Narten (ibm), Yakov Rekhter (cisco), Pyda Srisuresh, Matt Holdrege (lucent), and Eliot Lear (cisco).

14. Author's Address

Tony Hain
Microsoft
One Microsoft Way
Redmond, Wa. USA

Phone: 1-425-703-6619
EMail: tonyhain@microsoft.com

Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

