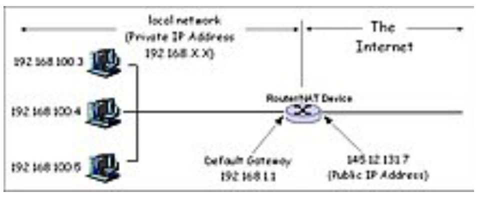
WIKIPEDIA

Network address translation

Network address translation (**NAT**) is a method of remapping one IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device.[1] The technique was originally used as a shortcut to avoid the need to readdress every host when a network was moved. It has become a popular and essential tool in conserving global address space in the face of IPv4 address exhaustion. One Internet-routable IP address of a NAT gateway can be used for an entire private network.



NAT for a private network

IP masquerading is a technique that hides an entire IP address space, usually consisting of private IP addresses, behind a single IP address in another, usually public address space. The hidden addresses are changed into a single (public) IP address as the source address of the outgoing IP packets so they appear as originating not from the hidden host but from the routing device itself. Because of the popularity of this technique to conserve IPv4 address space, the term *NAT* has become virtually synonymous with IP masquerading.

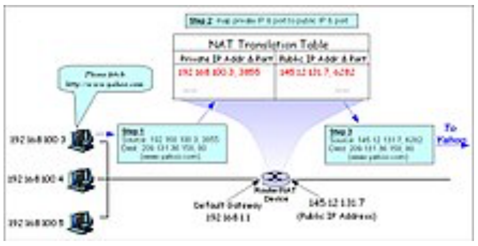
As network address translation modifies the IP address information in packets, NAT implementations may vary in their specific behavior in various addressing cases and their effect on network traffic. The specifics of NAT behavior are not commonly documented by vendors of equipment containing NAT implementations.[2]

Contents

Basic NAT

The simplest type of NAT provides a one-to-one translation of IP addresses. RFC 2663 refers to this type of NAT as *basic NAT*; it is also called a *one-to-one NAT*. In this type of NAT, only the IP addresses, IP header checksum and any higher-level checksums that include the IP address are changed. Basic NATs can be used to interconnect two IP networks that have incompatible addressing.

One-to-many NAT



Network address mapping

The majority of NATs map multiple private hosts to one publicly exposed IP address. In a typical configuration, a local network uses one of the designated *private* IP address subnets (RFC 1918). A router on that network has a private address in that address space. The router is also connected to the Internet with a *public* address assigned by an Internet service provider. As traffic passes from the local network to the Internet, the source address in each packet is translated on the fly from a private address to the public address. The router tracks basic data about each active connection (particularly the destination address and port). When a reply returns to the router, it uses the connection tracking data it stored during the outbound phase to determine the private address on the internal network to which to forward the reply.

All IP packets have a source IP address and a destination IP address. Typically packets passing from the private network to the public network will have their source address modified, while packets passing from the public network back to the private network will have their destination address modified. To avoid ambiguity in how replies are translated, further modifications to the packets are required. The vast bulk of Internet traffic uses Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). For these protocols the port numbers are changed so that the combination of IP address and port information on the returned packet can be unambiguously mapped to the corresponding private network destination. RFC 2663 uses the term *network address and port translation* (NAPT) for this type of NAT. Other names include *port address translation* (PAT), *IP masquerading*, *NAT overload* and *many-to-one NAT*. This is the most common type of NAT and has become synonymous with the term "NAT" in common usage.

This method enables communication through the router only when the conversation originates in the private network since the initial originating transmission is what establishes the required information in the translation tables. A web browser in the masqueraded network can, for example, browse a website outside, but a web browser outside cannot browse a website hosted within the masqueraded network.[a] Protocols not based on TCP and UDP require other translation techniques.

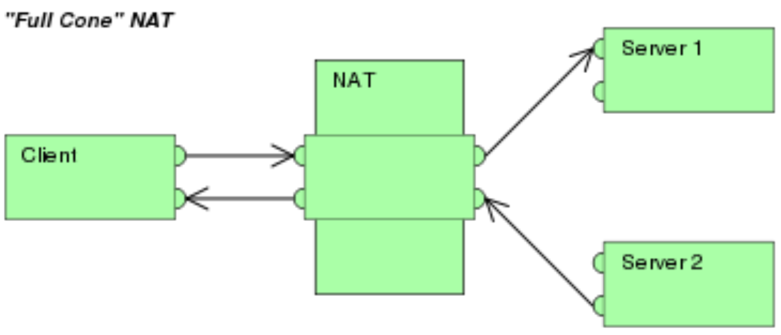
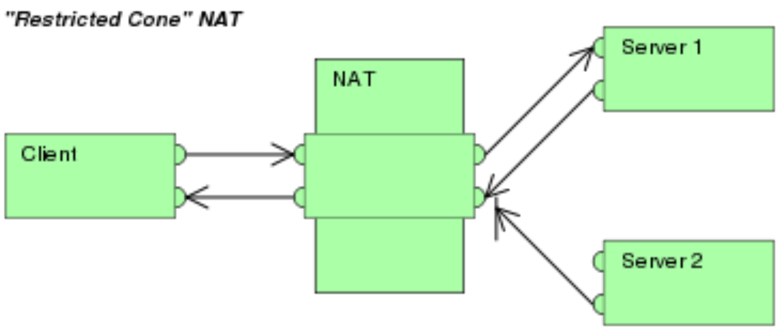
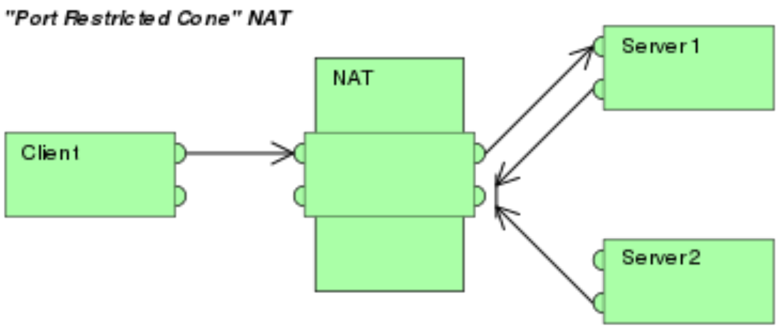
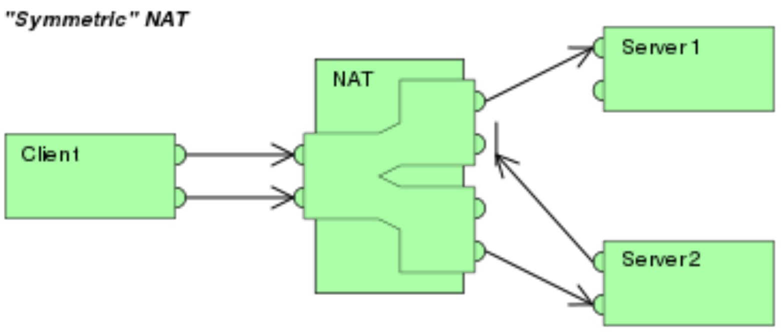
One of the additional benefits of one-to-many NAT is that it is a practical solution to IPv4 address exhaustion. Even large networks can be connected to the Internet using a single public IP address.[b]

Methods of translation

There are several ways of implementing network address and port translation. In some application protocols that use IP address information, the application running on a node in the masqueraded network needs to determine the external address of the NAT, i.e., the address that its communication peers detect, and, furthermore, often needs to examine and categorize the type of mapping in use. Usually this is done because it is desired to set up a direct communication path (either to save the cost of taking the data via a server or to improve performance) between two clients both of which are behind separate NATs.

For this purpose, the *Simple traversal of UDP over NATs* (STUN) protocol was developed (RFC 3489, March 2003). It classified NAT implementation as *full-cone NAT*, *(address) restricted-cone NAT*, *port-restricted cone NAT* or *symmetric NAT* and proposed a methodology for testing a device accordingly. However, these procedures have since been deprecated from standards status, as the methods are inadequate to correctly assess many devices. New methods have been standardized in RFC 5389 (October 2008) and the STUN acronym now represents the new title of the specification: *Session Traversal Utilities for NAT*.

NAT implementation classifications

Full-cone NAT, also known as *one-to-one NAT* ▪ Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort. ▪ *Any external host* can send packets to iAddr:iPort by sending packets to eAddr:ePort.	
(Address)-restricted-cone NAT ▪ Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort. ▪ An external host (*hAddr:any*) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:*any*. "Any" means the port number doesn't matter.	 "Restricted Cone" NAT NAT Client Server 1 Server 2
Port-restricted cone NAT Like an address restricted cone NAT, but the restriction includes port numbers. ▪ Once an internal address (iAddr:iPort) is mapped to an external address (eAddr:ePort), any packets from iAddr:iPort are sent through eAddr:ePort. ▪ An external host (*hAddr:hPort*) can send packets to iAddr:iPort by sending packets to eAddr:ePort only if iAddr:iPort has previously sent a packet to hAddr:hPort.	 "Port Restricted Cone" NAT NAT Client Server 1 Server2
Symmetric NAT ▪ Each request from the same internal IP address and port to a specific destination IP address and port is mapped to a unique external source IP address and port; if the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used. ▪ Only an external host that receives a packet from an internal host can send a packet back.	

This terminology has been the source of much confusion, as it has proven inadequate at describing real-life NAT behavior. Many NAT implementations combine these types, and it is, therefore, better to refer to specific individual NAT behaviors instead of using the Cone/Symmetric terminology. RFC 4787 attempts to alleviate this issue by introducing standardized terminology for observed behaviors. For the first bullet in each row of the above table, the RFC would characterize Full-Cone, Restricted-Cone, and Port-Restricted Cone NATs as having an *Endpoint-Independent Mapping*, whereas it would characterize a Symmetric NAT as having an *Address- and Port-Dependent Mapping*. For the second bullet in each row of the above table, RFC 4787 would also label Full-Cone NAT as having an *Endpoint-Independent Filtering*, Restricted-Cone NAT as having an *Address-Dependent Filtering*, Port-Restricted Cone NAT as having an *Address and Port-Dependent Filtering*, and Symmetric NAT as having either an *Address-Dependent Filtering* or *Address and Port-Dependent Filtering*. There are other classifications of NAT behavior mentioned, such as whether they preserve ports, when and how mappings are refreshed, whether external mappings can be used by internal hosts (i.e., its hairpinning behavior), and the level of determinism NATs exhibit when applying all these rules.[2]

Especially, most NATs combine *symmetric NAT* for outgoing connections with *static port mapping*, where incoming packets addressed to the external address and port are redirected to a specific internal address and port. Some products can redirect packets to several internal hosts, *e.g.*, to divide the load between a few servers. However, this introduces problems with more sophisticated communications that have many interconnected packets, and thus is rarely used.

Related techniques

IEEE[3] Reverse Address and Port Translation (RAPT, or RAT) allows a host whose real IP address is changing from time to time to remain reachable as a server via a fixed home IP address. In principle, this should allow setting up servers on DHCP-run networks. While not a perfect mobility solution, RAPT together with upcoming protocols like DHCP-DDNS, it may end up becoming another useful tool in the network admin's arsenal.

Cisco *RAPT* implementation is port address translation (PAT) or NAT overloading, and maps multiple private IP addresses to a single public IP address. Multiple addresses can be mapped to a single address because each private address is tracked by a port number. PAT uses unique source port numbers on the inside global IP address to distinguish between translations. The port numbers are 16-bit integers. The total number of internal addresses that can be translated to one external address could theoretically be as high as 65,536 per IP address. Realistically, the number of ports that can be assigned a single IP address is around 4000. PAT attempts to preserve the original source port. If this source port is already used, PAT assigns the first available port number starting from the beginning of the appropriate port group 0–511, 512–1023, or 1024–65535. When there are no more ports available and there is more than one external IP address configured, PAT moves to the next IP address to try to allocate the original source port again. This process continues until it runs out of available ports and external IP addresses.

Mapping of Address and Port is a Cisco proposal which combines A+P port address translation with tunneling of the IPv4 packets over an ISP provider's internal IPv6 network. In effect, it is an (almost) stateless alternative to Carrier Grade NAT and DS-Lite that pushes the IPv4 IP address/port translation function (and therefore the maintenance of NAT state) entirely into the existing customer premises equipment NAT implementation. Thus avoiding the NAT444 and statefulness problems of Carrier Grade NAT, and also provides a transition mechanism for the deployment of native IPv6 at the same time with very little added complexity.

Type of NAT and NAT traversal, role of port preservation for TCP

The NAT traversal problem arises when two peers behind distinct NAT try to communicate. One way to solve this problem is to use port forwarding, another way is to use various NAT traversal techniques. The most popular technique for TCP NAT traversal is TCP hole punching, which requires the NAT to follow the *port preservation* design for TCP, as explained below.

Many NAT implementations follow the *port preservation* design for TCP: for a given outgoing TCP communication, they use the same values as internal and external port numbers. NAT *port preservation* for outgoing TCP connections is crucial for TCP NAT traversal, because as TCP requires that one port can only be used for one communication at a time, programs bind distinct TCP sockets to ephemeral ports for each TCP communication, rendering NAT port prediction impossible for TCP.[2]

On the other hand, for UDP, NATs do not need to have *port preservation*. Indeed, multiple UDP communications (each with a distinct endpoint) can occur on the same source port, and applications usually reuse the same UDP socket to send packets to distinct hosts. This makes port prediction straightforward, as it is the same source port for each packet.

Furthermore, *port preservation* in NAT for TCP allows P2P protocols to offer less complexity and less latency because there is no need to use a third party (like STUN) to discover the NAT port since the application itself already knows the NAT port.[2][4]

However, if two internal hosts attempt to communicate with the same external host using the same port number, the external port number used by the second host is chosen at random. Such NAT is sometimes perceived as *(address) restricted cone NAT* and other times as *symmetric NAT*.

As of 2006, roughly 70% of the clients in P2P networks employed some form of NAT.[5]

Implementation

Establishing two-way communication

Every TCP and UDP packet contains a source IP address and source port number as well as a destination IP address and destination port number. The IP address/port number pair forms a socket. In particular, the source IP address and source port number form the source socket.

For publicly accessible services such as web servers and mail servers the port number is important. For example, port 80 connects to the web server software and port 25 to a mail server's SMTP daemon. The IP address of a public server is also important, similar in global uniqueness to a postal address or telephone number. Both IP address and port number must be correctly known by all hosts wishing to successfully communicate.

Private IP addresses as described in RFC 1918 are significant only on private networks where they are used, which is also true for host ports. Ports are unique endpoints of communication on a host, so a connection through the NAT device is maintained by the combined mapping of port and IP address.

PAT (Port Address Translation) resolves conflicts that would arise through two different hosts using the same source port number to establish unique connections at the same time.

Telephone number extension analogy

A NAT device is similar to a phone system at an office that has one public telephone number and multiple extensions. Outbound phone calls made from the office all appear to come from the same telephone number. However, an incoming call that does not specify an extension cannot be transferred to an individual inside the office. In this scenario, the office is a private LAN, the main phone number is the public IP address, and the individual extensions are unique port numbers.[6]

Translation of the endpoint

With NAT, all communications sent to external hosts actually contain the *external* IP address and port information of the NAT device instead of internal host IP addresses or port numbers.

- When a computer on the private (internal) network sends an IPv4 packet to the external network, the NAT device replaces the internal IP address in the source field of the packet header (*sender's address*) with the external IP address of the NAT device. PAT may then assign the connection a port number from a pool of available ports, inserting this port number in the source port field (much like the *post office box number*), and forwards the packet to the external network. The NAT device then makes an entry in a translation table containing the internal IP address, original source port, and the translated source port. Subsequent packets from the same connection are translated to the same port number.
- The computer receiving a packet that has undergone NAT establishes a connection to the port and IP address specified in the altered packet, oblivious to the fact that the supplied address is being translated (analogous to using a *post office box number*).
- A packet coming from the external network is mapped to a corresponding internal IP address and port number from the translation table, replacing the external IP address and port number in the incoming packet header (similar to the translation from *post office box number* to *street address*). The packet is then forwarded over the inside network. Otherwise, if the destination port number of the incoming packet is not found in the translation table, the packet is dropped or rejected because the PAT device doesn't know where to send it.

NAT only translates IP addresses and ports of its internal hosts, hiding the true endpoint of an internal host on a private network.

Visibility of operation

NAT operation is typically transparent to both the internal and external hosts.

Typically the internal host is aware of the true IP address and TCP or UDP port of the external host. Typically the NAT device may function as the default gateway for the internal host. However the external host is only aware of the public IP address for the NAT device and the particular port being used to communicate on behalf of a specific internal host.

Issues and limitations

Hosts behind NAT-enabled routers do not have end-to-end connectivity and cannot participate in some Internet protocols. Services that require the initiation of TCP connections from the outside network, or stateless protocols such as those using UDP, can be disrupted. Unless the NAT router makes a specific effort to support such protocols, incoming packets cannot reach their destination. Some protocols can accommodate one instance of NAT between participating hosts ("passive mode" FTP, for example), sometimes with the assistance of an application-level gateway (see below), but fail when both systems are separated from the Internet by NAT. Use of NAT also complicates tunneling protocols such as IPsec because NAT modifies values in the headers which interfere with the integrity checks done by IPsec and other tunneling protocols.

End-to-end connectivity has been a core principle of the Internet, supported for example by the Internet Architecture Board. Current Internet architectural documents observe that NAT is a violation of the end-to-end principle, but that NAT does have a valid role in careful design.[7] There is considerably more concern with the use of IPv6 NAT, and many IPv6 architects believe IPv6 was intended to remove the need for NAT.[8]

An implementation that only tracks ports can be quickly depleted by internal applications that use multiple simultaneous connections (such as an HTTP request for a web page with many embedded objects). This problem can be mitigated by tracking the destination IP address in addition to the port (thus sharing a single local port with many remote hosts), at the expense of implementation complexity and CPU/memory resources of the translation device.

Because the internal addresses are all disguised behind one publicly accessible address, it is impossible for external hosts to initiate a connection to a particular internal host without special configuration on the firewall to forward connections to a particular port. Applications such as VOIP, videoconferencing, and other peer-to-peer applications must use NAT traversal techniques to function.

NAT and TCP/UDP

"Pure NAT", operating on IP alone, may or may not correctly parse protocols that are totally concerned with IP information, such as ICMP, depending on whether the payload is interpreted by a host on the "inside" or "outside" of translation. As soon as the protocol stack is traversed, even with such basic protocols as TCP and UDP, the protocols will break unless NAT takes action beyond the network layer.

IP packets have a checksum in each packet header, which provides error detection only for the header. IP datagrams may become fragmented and it is necessary for a NAT to reassemble these fragments to allow correct recalculation of higher-level checksums and correct tracking of which packets belong to which connection.

The major transport layer protocols, TCP and UDP, have a checksum that covers all the data they carry, as well as the TCP/UDP header, plus a "pseudo-header" that contains the source and destination IP addresses of the packet carrying the TCP/UDP header. For an originating NAT to pass TCP or UDP successfully, it must recompute the TCP/UDP header checksum based on the translated IP addresses, not the original ones, and put that checksum into the TCP/UDP header of the first packet of the fragmented set of packets. The receiving NAT must recompute the IP checksum on every packet it passes to the destination host, and also recognize and recompute the TCP/UDP header using the retranslated addresses and pseudo-header. This is not a completely solved problem. One solution is for the receiving NAT to reassemble the entire segment and then recompute a checksum calculated across all packets.

The originating host may perform Maximum transmission unit (MTU) path discovery to determine the packet size that can be transmitted without fragmentation, and then set the *don't fragment* (DF) bit in the appropriate packet header field. Of course, this is only a one-way solution, because the responding host can send packets of any size, which may be fragmented before reaching the NAT.

DNAT

Destination network address translation (DNAT) is a technique for transparently changing the destination IP address of an end route packet and performing the inverse function for any replies. Any router situated between two endpoints can perform this transformation of the packet.

DNAT is commonly used to publish a service located in a private network on a publicly accessible IP address. This use of DNAT is also called port forwarding, or DMZ when used on an entire server, which becomes exposed to the WAN, becoming analogous to an undefended military demilitarised zone (DMZ).

SNAT

The meaning of the term *SNAT* varies by vendor.[9] Many vendors have proprietary definitions for *SNAT*:

- *source NAT* is the common expansion, as the counterpart of *destination NAT* (*DNAT*)
- *stateful NAT* is used by Cisco Systems[10]
- *static NAT* is used by WatchGuard[11]
- *secure NAT* is used by F5 Networks and by Microsoft (in regard to the ISA Server)

Microsoft's Secure network address translation (SNAT) is part of Microsoft's Internet Security and Acceleration Server and is an extension to the NAT driver built into Microsoft Windows Server. It provides connection tracking and filtering for the additional network connections needed for the FTP, ICMP, H.323, and PPTP protocols as well as the ability to configure a transparent HTTP proxy server.

Dynamic network address translation

Dynamic NAT, just like static NAT, is not common in smaller networks but is found within larger corporations with complex networks. The way dynamic NAT differs from static NAT is that where static NAT provides a one-to-one internal to public static IP address mapping, dynamic NAT usually uses a *group* of available public IP addresses.

NAT hairpinning

NAT hairpinning, also known as *NAT loopback* or *NAT reflection*,[12] is a feature in many consumer routers[13] that permits the access of a service via the public IP address from inside the local network. This eliminates the need for using separate domain name resolution for hosts inside the network than for the public network for a website.

The following describes an example network:

- Public address: *203.0.113.1*. This is the address of the WAN interface on the router.
- Internal address of router: *192.168.1.1*
- Address of the server: *192.168.1.2*
- Address of a local computer: *192.168.1.100*

If a packet is sent to the public address by a computer at *192.168.1.100*, the packet would normally be routed to the default gateway (the router), unless an explicit route is set in the computer's routing tables. A router with the NAT loopback feature detects that *203.0.113.1* is the address of its WAN interface, and treats the packet as if coming from that interface. It determines the destination for that packet, based on DNAT (port forwarding) rules for the destination. If the data were sent to port 80 and a DNAT rule exists for port 80 directed to *192.168.1.2*, then the host at that address receives the packet.

If no applicable DNAT rule is available, the router drops the packet. An ICMP Destination Unreachable reply may be sent. If any DNAT rules were present, address translation is still in effect; the router still rewrites the source IP address in the packet. The local computer (*192.168.1.100*) sends the packet as coming from *192.168.1.100*, but the server (*192.168.1.2*) receives it as coming from *203.0.113.1*. When the server replies, the process is identical as for an external sender. Thus, two-way communication is possible between hosts inside the LAN network via the public IP address.

NAT in IPv6

Network address translation is not commonly used in IPv6, because one of the design goals of IPv6 is to restore end-to-end network connectivity.[14] NAT loopback is not commonly needed. Although still possible, the large addressing space of IPv6 obviates the need to conserve addresses and every device can be given a unique globally routable address. That being said, using unique local addresses in combination with network prefix translation can achieve similar results.

Applications affected by NAT

Some application layer protocols (such as FTP and SIP) send explicit network addresses within their application data. FTP in active mode, for example, uses separate connections for control traffic (commands) and for data traffic (file contents). When requesting a file transfer, the host making the request identifies the corresponding data connection by its network layer and transport layer addresses. If the host making the request lies behind a simple NAT firewall, the translation of the IP address and/or TCP port number makes the information received by the server invalid. The Session Initiation Protocol (SIP) controls many Voice over IP (VoIP) calls, and suffers the same problem. SIP and SDP may use multiple ports to set up a connection and transmit voice stream via RTP. IP addresses and port numbers are encoded in the payload data and must be known before the traversal of NATs. Without special techniques, such as STUN, NAT behavior is unpredictable and communications may fail.

Application Layer Gateway (ALG) software or hardware may correct these problems. An ALG software module running on a NAT firewall device updates any payload data made invalid by address translation. ALGs need to understand the higher-layer protocol that they need to fix, and so each protocol with this problem requires a separate ALG. For example, on many Linux systems there are kernel modules called *connection trackers* that serve to implement ALGs. However, ALG does not work if the control channel is encrypted (e.g. FTPS).

Another possible solution to this problem is to use NAT traversal techniques using protocols such as STUN or ICE, or proprietary approaches in a session border controller. NAT traversal is possible in both TCP- and UDP-based applications, but the UDP-based technique is simpler, more widely understood, and more compatible with legacy NATs. In either case, the high-level protocol must be designed with NAT traversal in mind, and it does not work reliably across symmetric NATs or other poorly behaved legacy NATs.

Other possibilities are UPnP Internet Gateway Device Protocol, NAT-PMP (NAT Port Mapping Protocol), or Port Control Protocol (PCP),[15] but these require the NAT device to implement that protocol.

Most traditional client–server protocols (FTP being the main exception), however, do not send layer 3 contact information and therefore do not require any special treatment by NATs. In fact, avoiding NAT complications is practically a requirement when designing new higher-layer protocols today (e.g. the use of SFTP instead of FTP).

NATs can also cause problems where IPsec encryption is applied and in cases where multiple devices such as SIP phones are located behind a NAT. Phones that encrypt their signaling with IPsec encapsulate the port information within an encrypted packet, meaning that NA(P)T devices cannot access and translate the port. In these cases the NA(P)T devices revert to simple NAT operation. This means that all traffic returning to the NAT is mapped onto one client, causing service to more than one client "behind" the NAT to fail. There are a couple of solutions to this problem: one is to use TLS, which operates at level 4 in the OSI Reference Model and therefore does not mask the port number; another is to encapsulate the IPsec within UDP – the latter being the solution chosen by TISPAN to achieve secure NAT traversal, or a NAT with "IPsec Passthru" support.

Interactive Connectivity Establishment is a NAT traversal technique that does not rely on ALG support.

The DNS protocol vulnerability announced by Dan Kaminsky on July 8, 2008 is indirectly affected by NAT port mapping. To avoid DNS server cache poisoning, it is highly desirable not to translate UDP source port numbers of outgoing DNS requests from a DNS server behind a firewall that implements NAT. The recommended workaround for the DNS vulnerability is to make all caching DNS servers use randomized UDP source ports. If the NAT function de-randomizes the UDP source ports, the DNS server becomes vulnerable.

Examples of NAT software

- Internet Connection Sharing (ICS): NAT & DHCP implementation included with Windows desktop operating systems
- IPFilter: included with (Open)Solaris, FreeBSD and NetBSD, available for many other Unix-like operating systems
- ipfirewall (ipfw): FreeBSD-native packet filter
- Netfilter with iptables/nftables: the Linux packet filter
- NPF: NetBSD-native Packet Filter
- PF: OpenBSD-native Packet Filter
- Routing and Remote Access Service: routing implementation included with Windows Server operating systems
- WinGate: third-party routing implementation for Windows

See also

- Anything In Anything (AYIYA) — IPv6 over IPv4 UDP, thus working IPv6 tunneling over most NATs
- Gateway (telecommunications)
- Internet Gateway Device Protocol (IGD) — UPnP NAT-traversal method
- Middlebox
- Port triggering
- Hairpinning
- Subnetwork
- Port (computer networking)
- Teredo tunneling — NAT traversal using IPv6
- Carrier-grade NAT – NAT behind NAT within ISP.

Notes

a. Most NAT devices today allow the network administrator to configure static translation table entries for connections from the external network to the internal masqueraded network. This feature is often referred to as *static NAT*. It may be implemented in two types: port forwarding which forwards traffic from a specific external port to an internal host on a specified port, and designation of a DMZ host which passes all traffic received on the external interface (on any port number) to an internal IP address while preserving the destination port. Both types may be available in the same NAT device.

b. The more common arrangement is having computers that require end-to-end connectivity supplied with a routable IP address, while having others that do not provide services to outside users behind NAT with only a few IP addresses used to enable Internet access.

References

1. *Network Protocols Handbook* (https://books.google.com/books?id=D_GrQa2ZcLwC) (2 ed.). Javvin Technologies Inc. 2005. p. 27. ISBN 9780974094526. Retrieved 2014-09-16.
2. François Audet; Cullen Jennings (January 2007). *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP* (https://tools.ietf.org/html/rfc4787). IETF. doi:10.17487/RFC4787 (https://doi.org/10.17487%2FRFC4787). RFC 4787.
3. Singh, R.; Tay, Y.C.; Teo, W.T.; Yeow, S.W. (1999). "RAT: A quick (and dirty?) push for mobility support" (http://ieeexplore.ieee.org/iel4/6056/16183/00749275.pdf) (PDF). *IEEE Xplore – Sign In*. pp. 32–40. CiteSeerX 10.1.1.40.461 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.461). doi:10.1109/MCSA.1999.749275 (https://doi.org/10.1109%2FMCSA.1999.749275). ISBN 978-0-7695-0025-6.
4. "Characterization and Measurement of TCP Traversal through NATs and Firewalls" (http://nutss.gforge.cis.cornell.edu/pub/imc05-tcpnat/). December 2006.
5. "Illuminating the shadows: Opportunistic network and web measurement" (https://web.archive.org/web/20100724011252/http://illuminati.coralcdn.org/stats/). December 2006. Archived from the original (http://illuminati.coralcdn.org/stats/) on 2010-07-24.
6. "The Audio over IP Instant Expert Guide" (http://www.tieline.com/Downloads/Audio-over-IP-Instant-Expert-Guide-v1.pdf) (PDF). Tieline. January 2010. Retrieved 2011-08-19.
7. R. Bush; and D. Meyer; RFC 3439, *Some Internet Architectural Guidelines and Philosophy* (http://www.ietf.org/rfc/rfc3439.txt), December 2002
8. G. Van de Velde *et al.*; RFC 4864, *Local Network Protection for IPv6* (http://tools.ietf.org/rfc/rfc4864.txt), May 2007
9. "K7820: Overview of SNAT features" (https://support.f5.com/csp/article/K7820). *AskF5*. August 28, 2007. Retrieved February 24, 2019.
10. "Enhanced IP Resiliency Using Cisco Stateful NAT" (https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-software-releases-12-2-t/prod_white_paper0900aecd8052870b.html). *Cisco*.
11. "Configuration" (https://www.watchguard.com/help/configuration-examples/nat_to_email_servers_configuration_example%20(en-US).pdf) (PDF). *www.watchguard.com*.
12. "What is NAT Reflection/NAT Loopback/NAT Hairpinning?" (http://www.nycnetworkers.com/real-world/nat-reflectionnat-loopbacknat-hairpinning/). NYC Networkers. 2014-11-09. Retrieved 2017-04-27.
13. "NAT Loopback Routers – OpenSim" (http://opensimulator.org/wiki/NAT_Loopback_Routers) (MediaWiki). OpenSimulator. 2013-10-21. Retrieved 2014-02-21.
14. Iljitsch van Beijnum (2008-07-23). "After staunch resistance, NAT may come to IPv6 after all" (https://arstechnica.com/uncategorized/2008/07/after-staunch-resistance-nat-may-come-to-ipv6-after-all/). *Ars Technica*. Retrieved

2014-04-24.

15. RFC 6887, *Port Control Protocol (PCP)*, Wing, Cheshire, Boucadair, Penno & Selkirk (April 2013)

External links

- NAT-Traversal Test and results (http://nattest.net.in.tum.de)
- Characterization of different TCP NATs (http://nutss.net/pub/imc05-tcpnat/) – Paper discussing the different types of NAT
- Anatomy: A Look Inside Network Address Translators – Volume 7, Issue 3, September 2004 (http://www.cisco.com/en/US/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html)
- Jeff Tyson, HowStuffWorks: *How Network Address Translation Works* (http://computer.howstuffworks.com/nat.htm/printable)
- Routing with NAT (http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=/rzajw/rzajwstatic.htm) (Part of the documentation for the IBM iSeries)
- Network Address Translation (NAT) FAQ (http://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html) – Cisco Systems

Retrieved from "https://en.wikipedia.org/w/index.php?title=Network_address_translation&oldid=924876531"