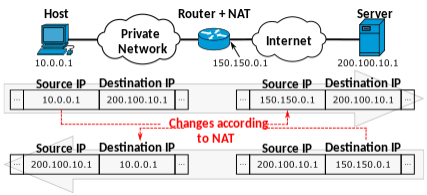
Network address translation

Network address translation (**NAT**) is a method of mapping an 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 to bypass the need to assign a new address to every host when a network was moved, or when the upstream Internet service provider was replaced, but could not route the network's address space. 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.[2]



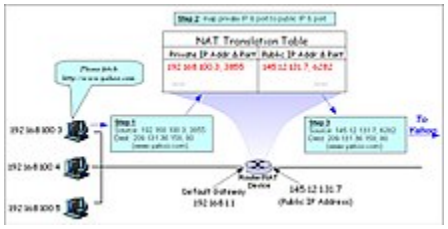
Network address translation between a private network and the Internet

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]

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 NAT can be used to interconnect two IP networks that have incompatible addressing.[2]

One-to-many NAT



Network address mapping

The majority of network address translators map multiple private hosts to one publicly exposed IP address.

Here is a typical configuration:

1. A local network uses one of the designated *private* IP address subnets (RFC 1918[3]).
2. The network has a router having both a private and a public address. The private address is used by the router for communicating with other devices in the private local network. The public address (typically assigned by an Internet service provider) is used by the router for communicating with the rest of the Internet.

3. As traffic passes from the network to the Internet, the router translates the source address in each packet from a private address to the router's own public address. The router tracks basic data about each active connection (particularly the destination address and port). When the router receives inbound traffic from the Internet, it uses the connection tracking data it stored during the outbound phase to determine to which private address (if any) it should forward the reply.[2]

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 (within the IP header) and port number (within the Transport Layer header) 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.[3] 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 allows 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. Thus a web browser within the private network would be able to browse websites that are outside the network, whereas web browsers outside the network would be unable to browse a website hosted within.[a] Protocols not based on TCP and UDP require other translation techniques.

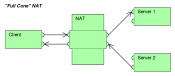
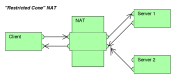
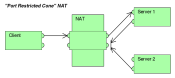
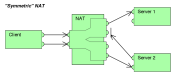
An additional benefit of one-to-many NAT is that it mitigates IPv4 address exhaustion by allowing entire networks to be connected to the Internet using a single public IP address.[b]

Methods of translation

Network address and port translation may be implemented in several ways. Some applications that use IP address information may need to determine the external address of a network address translator. This is the address that its communication peers in the external network detect. Furthermore, it may be necessary to examine and categorize the type of mapping in use, for example when it is desired to set up a direct communication path between two clients both of which are behind separate NAT gateways.

For this purpose, RFC 3489 specified a protocol called *Simple Traversal of UDP over NATs* (STUN) in 2003. It classified NAT implementations 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. RFC 5389 standardized new methods in 2008 and the acronym *STUN* 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.	
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.	
Symmetric NAT • The combination of one internal IP address plus a destination IP address and port is mapped to a single 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.	

Many NAT implementations combine these types, so it is better to refer to specific individual NAT behavior instead of using the Cone/Symmetric terminology. RFC 4787 attempts to alleviate confusion 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*. Other classifications of NAT behavior mentioned in the RFC include 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] Specifically, 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.

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

The NAT traversal problem arises when peers behind different NATs 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.

TCP hole punching requires the NAT to follow the *port preservation* design for TCP. For a given outgoing TCP communication, the same port numbers are used on both sides of the NAT. NAT port preservation for outgoing TCP connections is crucial for TCP NAT traversal because, under TCP, one port can only be used for one communication at a time, so 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 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.

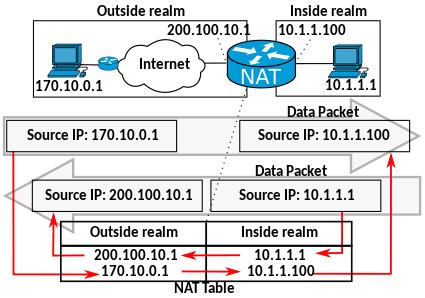
Furthermore, port preservation in NAT for TCP allows P2P protocols to offer less complexity and less latency because there is no need to use a third party (like STUN) to discover the NAT port since the application itself already knows the NAT port.[2][4]

However, if two internal hosts attempt to communicate with the same external host using the same port number, the NAT may attempt to use a different external IP address for the second connection or may need to forgo port preservation and remap the port.[2]:9

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

Implementation

Establishing two-way communication



In bidirectional NAT the session can be established both from inside and outside realms.

Every TCP and UDP packet contains a source port number and a destination port number. Each of those packets is encapsulated in an IP packet, whose IP header contains a source IP address and a destination IP address. The IP address/protocol/port number triple defines an association with a network socket.

For publicly accessible services such as web and mail servers the port number is important. For example, port 80 connects through a socket 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 usable only on private networks not directly connected to the internet. Ports are endpoints of communication unique to that host, so a connection through the NAT device is maintained by the combined mapping of port and IP address. A private address on the inside of the NAT is mapped to an external public address. Port address translation (PAT) resolves conflicts that arise when multiple hosts happen to use the same source port number to establish different external 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 automatically 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 process

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

When a computer on the private (internal) network sends an IP packet to the external network, the NAT device replaces the internal source IP address in the packet header 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. The packet is then forwarded 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 internal source IP address and port number are translated to the same external source IP address and 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.

Upon receiving a packet from the external network, the NAT device searches the translation table based on the destination port in the packet header. If a match is found, the destination IP address and port number is replaced with the values found in the table and the packet is forwarded to 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.

Visibility of operation

NAT operation is typically transparent to both the internal and external hosts. The NAT device may function as the default gateway for the internal host which is typically aware of the true IP address and TCP or UDP port of the external 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.

Applications

Routing

Network address translation can be used to mitigate IP address overlap.[7][8] Address overlap occurs when hosts in different networks with the same IP address space try to reach the same destination host. This is most often a misconfiguration and may result from the merger of two networks or subnets, especially when using RFC 1918 private network addressing. The destination host experiences traffic apparently arriving from the same network, and intermediate routers have no way to determine where reply traffic should be sent to. The solution is either renumbering to eliminate overlap or network address translation.

Load balancing

In client–server applications, load balancers forward client requests to a set of server computers to manage the workload of each server. Network address translation may be used to map a representative IP address of the server cluster to specific hosts that service the request.[9][10][11][12]

Related techniques

IEEE Reverse Address and Port Translation (RAPT or RAT) allows a host whose real IP address changes from time to time to remain reachable as a server via a fixed home IP address.[13] Cisco's RAPT implementation is 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.[c] 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 that combines Address plus Port 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 address/port translation function (and 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.

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 that use 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 § Applications affected by NAT), but fail when both systems are separated from the internet by NAT. The 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.[14] 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.[15]

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. This additional tracking increases implementation complexity and computing resources at the translation device.

Because the internal addresses are all disguised behind one publicly accessible address, it is impossible for external hosts to directly initiate a connection to a particular internal host.

Applications such as VOIP, videoconferencing, and other peer-to-peer applications must use NAT traversal techniques to function.

Fragmentation and checksums

Pure NAT, operating on IP alone, may or may not correctly parse protocols with payloads containing information about IP, such as ICMP. This depends on whether the payload is interpreted by a host on the *inside* or *outside* of the translation. Basic protocols as TCP and UDP cannot function properly 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.

TCP and UDP, have a checksum that covers all the data they carry, as well as the TCP or UDP header, plus a *pseudo-header* that contains the source and destination IP addresses of the packet carrying the TCP or UDP header. For an originating NAT to pass TCP or UDP successfully, it must recompute the TCP or UDP header checksum based on the translated IP addresses, not the original ones, and put that checksum into the TCP or UDP header of the first packet of the fragmented set of packets.

Alternatively, the originating host may perform path MTU 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. 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 a routed 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 demilitarized zone (DMZ).

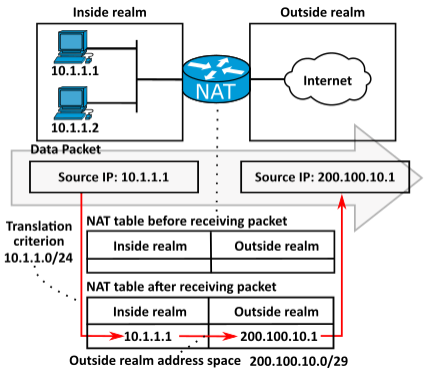
SNAT

The meaning of the term *SNAT* varies by vendor:[16][17][18]

- *source NAT* is a common expansion and is the counterpart of *destination NAT* (*DNAT*). This is used to describe one-to-many NAT; NAT for outgoing connections to public services.
- *stateful NAT* is used by Cisco Systems[19]
- *static NAT* is used by WatchGuard[20]
- *secure NAT* is used by F5 Networks[21] and by Microsoft (in regard to the ISA Server)

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



How dynamic NAT works.

Dynamic NAT, just like static NAT, is not common in smaller networks but is found within larger corporations with complex networks. Where static NAT provides a one-to-one internal to public static IP address mapping, dynamic NAT uses a *group* of public IP addresses.[22][23]

NAT hairpinning

NAT hairpinning, also known as **NAT loopback** or **NAT reflection**,[24] is a feature in many consumer routers[25] where a machine on the LAN is able to access another machine on the LAN via the external IP address of the LAN/router (with port forwarding set up on the router to direct requests to the appropriate machine on the LAN). This notion is officially described in 2008, RFC 5128 (https://datatracker.ietf.org/doc/html/rfc5128) .

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 *203.0.113.1* by a computer at *192.168.1.100*, the packet would normally be routed to the default gateway (the router)[d] 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 to 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.[26] The large addressing space of IPv6 obviates the need to conserve addresses and every device can be given a unique globally

routable address. Use of unique local addresses in combination with network prefix translation can achieve results similar to NAT.

Applications affected by NAT

Some application layer protocols, such as File Transfer Protocol (FTP) and Session Initiation Protocol (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 or TCP port number makes the information received by the server invalid. SIP commonly controls voice over IP calls, and suffers the same problem. SIP and its accompanying Session Description Protocol may use multiple ports to set up a connection and transmit voice stream via Real-time Transport Protocol. 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 cannot work if the protocol data is encrypted.

Another possible solution to this problem is to use NAT traversal techniques using protocols such as STUN or Interactive Connectivity Establishment (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 Internet Gateway Device Protocol, NAT Port Mapping Protocol (NAT-PMP), or Port Control Protocol (PCP),[27] but these require the NAT device to implement that protocol.

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

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 NAT devices cannot access and translate the port. In these cases the NAT 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 layer 4 and 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; another is to use a session border controller to help traverse the NAT.

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 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

Wikimedia Commons has media related to ***Network Address Translation***.

- Anything In Anything (AYIYA) – IPv6 over IPv4 UDP, thus working IPv6 tunneling over most NATs
- Gateway (telecommunications) – Connection between two network systems
- Internet Gateway Device Protocol (IGD) – UPnP NAT-traversal method
- Middlebox – Intermediary box on the data path between a source host and destination host
- Port triggering – NAT traversal mechanism
- Hairpinning
- Subnetwork – Logical subdivision of an IP network
- 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.*

c. *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.*

d. *Unless an explicit route is set in the computer's routing tables.*

e. *This issue can be avoided by using SFTP instead of FTP*

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://datatracker.ietf.org/doc/html/rfc4787)* . *IETF*. *doi:10.17487/RFC4787 (https://doi.org/10.17487%2FRFC4787)* . *RFC 4787 (https://datatracker.ietf.org/doc/html/rfc4787)* .
3. Wing, Dan (2010-07-01). *"Network Address Translation: Extending the Internet Address Space" (http://ieeexplore.ieee.org/document/5496805/)* . IEEE Internet Computing. **14** (4): 66–70. *doi:10.1109/MIC.2010.96 (https://doi.org/10.1109%2FMIC.2010.96)* . *ISSN 1089-7801 (https://www.worldcat.org/issn/1089-7801)* .
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" (https://web.archive.org/web/20111008014142/http://www.tieline.com/Downloads/Audio-over-IP-Instant-Expert-Guide-v1.pdf)* (PDF). Tieline. January 2010. Archived from *the original (http://www.tieline.com/Downloads/Audio-over-IP-Instant-Expert-Guide-v1.pdf)* (PDF) on 2011-10-08. Retrieved 2011-08-19.
7. *"Using NAT in Overlapping Networks" (https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/13774-3.html)* . August 2005.
8. *"VPNs with Overlapping Subnets Problem Scenario" (https://www.juniper.net/documentation/en_US/release-independent/nce/topics/concept/lan2lan-vpn-jseries-srx-series-overview.html)* . September 2017.
9. Srisuresh, Pyda; Gan, Der-Hwa (August 1998). *"Load Sharing using IP Network Address Translation" (https://tools.ietf.org/html/rfc2391)* . *doi:10.17487/RFC2391 (https://doi.org/10.17487%2FRFC2391)* .
10. *"What Is Layer 4 Load Balancing?" (https://www.nginx.com/resources/glossary/layer-4-load-balancing)* . June 2020.
11. *"What is load balancing?" (https://nfware.com/blog-what-is-load-balancing)* . November 2018.
12. *"Configure Server Load Balancing Using Dynamic NAT" (https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/200608-Server-Load-Balancing-Using-Dynamic-NAT.html)* . June 2018.

13. *Singh, R.; Tay, Y.C.; Teo, W.T.; Yeow, S.W. (1999). "RAT: A quick (and dirty?) push for mobility support". Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications. 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. S2CID 7657883 (https://api.semanticscholar.org/CorpusID:7657883) .*

14. *Bush, R.; Meyer, D. (2002). Some Internet Architectural Guidelines and Philosophy (https://datatracker.ietf.org/doc/html/rfc3439) . IETF. doi:10.17487/RFC3439 (https://doi.org/10.17487%2FRFC3439) . RFC 3439 (https://datatracker.ietf.org/doc/html/rfc3439) .*

15. *Velde, G. Van de; Hain, T.; Droms, R.; Carpenter, B.; Klein, E. (2007). Local Network Protection for IPv6 (https://datatracker.ietf.org/doc/html/rfc4864) . IETF. doi:10.17487/RFC4864 (https://doi.org/10.17487%2FRFC4864) . RFC 4864 (https://datatracker.ietf.org/doc/html/rfc4864) .*

16. *"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.*

17. *"Use NAT for Public Accessto Servers with Private IP Addresses on the Private Network (WatchGuard configuration example)" (https://www.watchguard.com/help/configuration-examples/nat_to_email_servers_configuration_example%20(en-US).pdf) (PDF). www.watchguard.com.*

18. *"K7820: Overview of SNAT features" (https://support.f5.com/csp/article/K7820) . AskF5. August 28, 2007. Retrieved February 24, 2019.*

19. *"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.*

20. *"Use NAT for Public Accessto Servers with Private IP Addresses on the Private Network (WatchGuard configuration example)" (https://www.watchguard.com/help/configuration-examples/nat_to_email_servers_configuration_example%20(en-US).pdf) (PDF). www.watchguard.com.*

21. *"K7820: Overview of SNAT features" (https://support.f5.com/csp/article/K7820) . AskF5. August 28, 2007. Retrieved February 24, 2019.*

22. *"Dynamic NAT" (https://study-ccna.com/dynamic-nat/) . 26 January 2016. Retrieved 2022-04-19.*

23. *"Dynamic NAT" (https://docs.oracle.com/cd/E19047-01/sunscreen32/806-6347/6jfa0g880/index.html) . Retrieved 2022-04-19.*

24. *"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.*

25. *"NAT Loopback Routers – OpenSim" (http://opensimulator.org/wiki/NAT_Loopback_Routers) (MediaWiki). OpenSimulator. 2013-10-21. Retrieved 2014-02-21.*

26. *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.*

27. *D. Wing, Ed; Cheshire, S.; Boucadair, M.; Penno, R.; Selkirk, P. (2013). Port Control Protocol (PCP) (https://datatracker.ietf.org/doc/html/rfc6887) . IETF. doi:10.17487/RFC6887 (https://doi.org/10.17487%2FRFC6887) . RFC 6887 (https://datatracker.ietf.org/doc/html/rfc6887) .*

External links

- NAT-Traversal Test and results (https://web.archive.org/web/20180817001123/http://nattest.net.in.tum.de/)
- Characterization of different TCP NATs (https://web.archive.org/web/20140811202730/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 (https://archive.today/20130103041130/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

Last edited 5 days ago by Kvng

WIKIPEDIA