IEEE distributed systems ONLINE
Expert-authored articles and resources
July 2002

From the Jul./Aug. issue of *IEEE Internet Computing*
Story location:
http://dsonline.computer.org/0207/departments/wp4icon.htm

Those Pesky NATs

Joseph D. Touch • Postel Center for Experimental Networking, USC/ISI

Whether buried deep inside ISPs or camouflaged as DSL routers, the *network address translator* has become a ubiquitous tool in the Internet landscape. NATs enable telco and cable operators to prevent commercial use of consumer accounts. They also let home users run open community access wireless networks off a single purchased account. It is what NATs disable, however, that makes them nefarious.

How NATs Work

A NAT makes a group of networked computers appear to the rest of the Internet as if they were a single computer, using a single address.[1-3] To the Internet, traffic from the computers behind a NAT appears to come directly from the NAT itself. To accomplish this feat, NATs assign local addresses to the systems they hide and translate packets between them and the Internet. NATs keep an internal table to track associations and guide this translation.

In the example shown in Figure 1, machine B behind NAT X sends packets to machine Y outside the NAT, on the Internet. B emits packets from port 5 to port 9 on Y, so the packet from B looks like [B.5 → Y.9]. When the packet arrives, the NAT translates both the source address and source port as if they were coming from a different port on X, for example [X.7 → Y.9], and adds a translation entry to its table [B.5 ↔ X.7]. The NAT keeps track of present and past entries to translate incoming and outgoing packets. It rewrites packets arriving at X.7 and sends them to B.5, then sends further packets coming from B.5 as if from X.7.

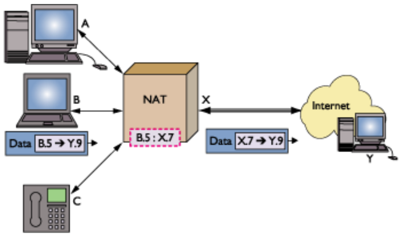

Figure 1. How a NAT works. By assigning local addresses to the systems they hide and translating packets between them and the Internet, NATs make traffic coming from computers behind the NAT appear as if it comes directly from the NAT Itself.

This simple set of rules lets a NAT hide a set of machines behind it, which can be useful when the number of available IP addresses is limited. It also has some obvious limits.[4-7]

- NATs work only when the translation works; they fail when addresses are used in ways the NAT is unaware of, such as when embedded in a packet payload or when encrypted (see Figure 2a).
- NATs work only for connections or associations that originate on hidden computers; machines behind a NAT cannot be called by systems out on the Internet (behind NATs or not); they can only call out (see Figure 2b).
- NATs assume a persistent association; that's what keeps the address translation table entry active. If a NAT keeps translation entries too long, it can run out of ports for translation; if it doesn't keep them long enough, the NAT can drop connections or break protocols. The result is the same as in Figure 2b: table entries are lost and incoming packets have no match.

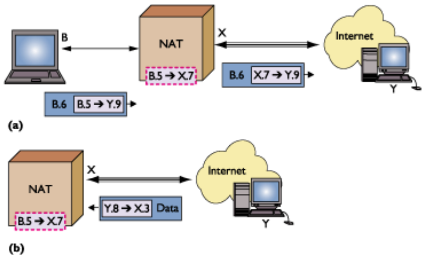

Figure 2. Problems NATs can cause. (a) In-band addresses fail if not translated, such as when callback addresses -- here B.6 -- are

not in the table. (b) Incoming calls fail, as when X.3 fails to match in the NAT table.

Causes of NAT Problems

There are many symptoms to NAT failures, as noted above, but only a few basic causes. Each major limitation associated with NATs inhibits some aspect of Internet use:

- in-band addresses (translation can't see inside data),
- servers (incoming calls fail), or
- long-idle connections (translation info is lost).

Each of these causes results in a translation failure. When translation fails, so does traffic through the NAT.

In-Band Addresses

In-band addresses are address (or port) numbers communicated inside a protocol's data payload. If the NAT's translator knows about the protocol and knows how to interpret the data, it can translate the internal addresses, too. NATs do not translate other protocols, which then break when used on computers behind NATs. These include video teleconferencing applications (which typically have a control channel that indicates address and port numbers for the separate video and audio channels), file transfer (FTP), and virtual private networks (Microsoft PPTP tunnels).

In-band addresses often serve to synchronize applications (as just discussed), but they are useful for authentication as well. In this latter case, the packet payload contains an encrypted message, indicating "this connection came from X." Again, the process breaks down when a NAT is introduced because it translates the source address X from the packet header, but cannot translate the "X" used in the encrypted body of the packet. When the receiver attempts to validate the message, an error occurs, because the outer header (translated X) won't match the inner address (untranslated X). This happens when the user has a machine-specific certificate, such as a Web certificate for running a secure Web server, or when using end-to-end security that includes the IP address in the key, as in IPsec. It also occurs when the user's IP address or DNS name is itself used for authentication, as with remote Unix services such as rsh, rlogin, and rcp (which use the .rhosts file), for X11 xhost, and for Web servers—for example, using .htaccess in Apache Web servers. Also, some systems use implicit in-band addresses, such as services that fail-over to other services: klogin to rlogin (secure and regular login), or IMAP to POP (email). In this case, the failover service is an implicit in-band address (here, a port number), which the NAT cannot translate.

Incoming Calls

Servers, by definition, must support incoming calls. When a packet arrives on a so-called "well-known port" (which identifies an application—for example, 80 is Web, 110 is IMAP email), it must be directed to the correct machine behind the NAT. For outgoing connections, the NAT knows the originating (hidden) machine and establishes a translation table association that allows returning packets to be properly converted. For incoming connections, there is no pre-existing translation table association; because it cannot know which system should get the packet, the NAT therefore drops the packet. The alternative is to route all unknown incoming calls to a single machine, known as a DMZ (demilitarized zone) server. Thus, for each service (incoming port number), there can be only one internal machine running that service behind the NAT.

Servers are not just for commercial companies. Indeed, user servers have become quite common in recent years as home users have started running Web servers for business purposes or to maintain things like picture archives. The former is one reason NATs are in such widespread use: to prohibit home users from usurping commercial fees for Web service from ISPs (see Why So Many NATs).

There are other server that users might not even know about. Some manufacturers, such as Compaq (now HP), provide Web servers as part of the system management architecture on their PCs. Such servers provide information on driver software versions to enable Compaq to suggest updates. Running a machine behind a NAT disables this feature, silently depriving computer owners of critical updates.

Other server-based architectures include IP telephony and video conferencing, Internet games, some kinds of messaging systems (instant messaging, Internet relay chat, and so on), peer-to-peer applications, and X-Windows (in which the display is a server and applications are clients).

Lost Translations

The third problem with NATs involves the pesky translation table and how long to keep entries. It might seem reasonable to add table entries when establishing connections and remove them when disconnecting. Only a subset of Internet protocols have connect and disconnect messages, however, and the disconnect messages can get lost. Other protocols require the use of timeouts. If the timeout is too short, the translation table drops entries before communication completes, and the NAT then either discards subsequent packets or misdirects them to other machines behind the NAT. If the timeout is too long, the port numbers of the NAT are held in use, and the NAT runs out of numbers for new translations, which causes new connections to fail.

In addition to this timeout issue, the NAT must maintain numerous protocol rules, such as when a port used for a connection to one host cannot be reused for a connection to another host for a few minutes. If this rule is violated, data sent on reliable channels (via TCP) can mistakenly appear on other reliable channels.

Why So Many NATs?

NATs have become so ubiquitous, in part, because ISPs use them internally to support their business model. Like many businesses, ISPs would like to charge business customers more than home users, particularly to subsidize infrastructure and provide competitive consumer prices. To charge different prices, providers need to offer different levels of service. To that end, ISPs decided early on that businesses needed to run Web servers and consumers probably didn't. NATs help enforce this model, in which consumers' uplink connections are aggregated to reduce costs, and businesses pay hefty fees (by a factor of five or more) for the privilege of running servers.

NATs to You?

To find out if you're behind a NAT, you can compare your computer's IP address to the address seen on the Internet. Your address will be visible in the configuration of your network interface (Ethernet, wireless card, and so forth). On Windows PCs, find a command window and type "ipconfig" (earlier versions use "winipcfg"); on Unix PCs, it's "ifconfig". To find your address as seen from the Internet, you need to find a remote site to help. Compaq has a Web page that shows the address it thinks you are connecting from (http://wwss1pro.compaq.com/support/tools). If your address from ipconfig doesn't match the externally visible result, you're behind at least one NAT at the externally visible address.

One alleged use of NATs, conservation of address space, can be secondary to this model. Because a NAT makes a set of hidden machines look like a single Internet machine, fewer Internet addresses can be used to support more machines. NATs do both—conserve addresses and prevent consumer servers. Dynamic IP address leases (using the dynamic host configuration protocol) can also defeat customer servers, and ISPs sometimes use these leases even when they don't use NATs. In either case, the ISP business model supports either practice (NATs or dynamic leases)—at the expense of breaking some user services.

NATs can be useful in limited environments. They do let users run multiple PCs on a home network, for instance, because the ISP sees only one host (the NAT) and thus charges for a single address. (Paradoxically, the same ISPs that promoted NATs because doing so helped their business models are now losing revenue as customers create open wireless community networks by sharing a single connection among dozens of machines, all hidden by a NAT.) Although not generally accepted as sufficient for real security, NATs can also thwart hackers by breaking incoming calls. Despite their beneficial uses, however, NATs are guaranteed to break something eventually.

Because a NAT can hide inside the ISP or local network infrastructure, it can be difficult to determine where a protocol fails and that a NAT is responsible. Also, an individual user can do little to overcome problems created by NATs. Proposed solutions—intelligent NATs that can be reprogrammed for new protocols, middle-box interfaces to let hidden computers find NATs and control them, and alternative translations that decouple end addresses from connection addresses (realm-specific IP [RSIP] and host identity payload protocol [HIP], for example)—still eventually encounter some of these problems in some way.

Conclusions

The best solution is to avoid NATs altogether and insist on real Internet addresses. Some ISPs will provide real IP addresses for a small additional charge. If yours does not, shop around. In the current version of IPv4, addresses might be in short supply if we all ask at once, but this shortage could provide the needed incentive for ISPs to step up support for IPv6, which has more than enough address space.

Also, don't be fooled by so-called DSL cable "routers"; they're just NATs in disguise. If you want a router, insist that it actually routes. You might not have a protocol that breaks under NATs now, but (to paraphrase Yoda), you will.

Joseph D. Touch is the director of the Postel Center for Experimental Networking at the Univ. of Southern California's Information Sciences Institute where he also serves as research associate professor. He has a PhD in computer and information sciences from the University of Pennsylvania. Contact him at touch@isi.edu; http://www.isi.edu/touch.

Resources

1. J. Tyson, "How Network Address Translation Works"
2. Vicomsoft NAT FAQ
3. Linksys NAT FAQ
4. L. Phifer, "The Trouble with NAT," Cisco's Internet Protocol Journal, Dec. 2000.
5. K. Moore, "Things that NATs Break"
6. M. Holdrege and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," Internet Engineering Task Force, RFC-3027, Jan. 2001.
7. T. Hain, "Architectural Implications of NAT," IETF RFC-2993, Nov. 2000.

DS Online ISSN: 1541-4922 • Feedback? Send comments to dsonline@computer.org.