Is the Internet Going NUTSS?

Paul Francis • *Cornell University*

For nearly 10 years now, the Internet Architecture Board (IAB) and Internet Engineering Task Force (IETF) have been telling us that the pool of available IP addresses will soon be exhausted, and that Internet growth will come to a grinding halt. They have heavily promoted their solution, IPv6, which the commercial world has all but ignored.

It is now becoming clear that IP address exhaustion is years off, at best. The primary reason for this is network address translation (NAT), the rogue technology that allows almost unlimited address reuse. Despite NAT's nagging technical problems that limit IP connectivity and make peer-to-peer (P2P) applications difficult to deploy, the commercial world has universally embraced the technology even as the IAB and IETF actively discourage its use.

If the commercial world hasn't accepted IPv6 so far, what will make it do so in the future? If (as I believe) the answer is "nothing," isn't it about time the IAB and IETF started hedging their bets and looking at more commercially acceptable solutions to the IP-connectivity problem? To better understand the answers to these questions, it is instructive to look at the history of IPv6, NAT, and the prevailing attitude toward these technologies.

Hitting the Road

A little over a decade ago, I sat on a committee organized by Vint Cerf that was chartered to deal with the anticipated shortage of IP addresses. (This was long enough ago that I didn't yet know the names Tim Berners-Lee or Marc Andreessen.) The creation of this committee, called the Road Group (for "routing and addressing"), kicked off a process that has yet to reach closure.

At the time, many of us feared that the pool of available IP addresses might be depleted within just a few years, and so we worked with a sense of urgency. Vint hoped that the Road Group would quickly produce a solution backed by the Internet community's most respected members. With this backing, the solution would quickly be vetted and approved by the IETF and become reality.

The Road Group was a partial success. It produced something called classless interdomain routing (CIDR),[1] which allowed blocks of addresses to be assigned with finer granularity. Before CIDR, blocks of IP addresses came in three sizes only: 256; 65,000; and 16 million (corresponding to the historic Class C, B, and A address types, respectively). An organization with only a few thousand computers might be given 65,000 addresses (Class B), thus wasting 95 percent of them. With CIDR, blocks of addresses could come in any power of two.

CIDR stopped the immediate hemorrhaging of addresses, but it did not deal with the more fundamental fact that a puny 32 bits could not supply enough addresses for an Internet of global proportion. For this much harder problem, the Road Group failed. The chance to redesign IP to our individual liking was too much to resist, and we could not agree on an approach.

Instead, a process began in the IETF by which anybody could put forward a proposal for the next-generation IP protocol (IPng). Several of us generated designs, and there was heated debate over the merits and shortcomings of each. To make a long story short, a couple of years later this process produced what we now call IPv6, which very nearly matched the original proposal from Steve Deering — more than anybody, the father of IPv6.

A Short-Term Solution?

During the same period (1992–1993), I invented NAT, which I thought at the time would be a short-term stopgap solution to the address depletion problem. NAT allows a group of hosts at a site to share a single IP address. The site hosts are given "private" (nonunique) addresses[2] that a NAT box placed between the site and its ISP translates dynamically to the site's global address(es). My original design mapped one global address to one private address at a time.[3] Later, this was enhanced with TCP/UDP port translation, which allows the NAT box to translate one global address to thousands of private addresses at a time.[4] (I don't know who thought of this enhancement.)

Many of the more influential IETF members hated NAT (and still do). This might surprise anyone who came into IP technology after 1995. At the time, the idea of not carrying the IP address intact end-to-end was shocking and disturbing. In those days, many systems did not have access to DNS, so the IP address was the only stable identifier you could rely on. Most people knew their own computer's IP address, and it was not unheard of to see it on a business card. Indeed, the most common reaction to NAT at the time was, "you can actually do that?" (This was typically followed by an emphatic "yuck!")

By 1995, it was pretty clear that NAT was a commercial success and not going away anytime soon. It was also clear that NAT was effective at preserving IP addresses. The IETF was faced with a choice between accommodating or discouraging NAT. Accommodation would have meant, at the least, ensuring that new IETF protocols operated correctly through NATs, and developing guidelines for predictable NAT operation. Furthermore, it could have meant solving NAT's primary shortcoming, which is that it does not generally allow incoming connectivity — connections established from the global Internet to the private site. (It is worth noting that NAT vendors successfully, though largely incorrectly, marketed this shortcoming as a security feature of NAT.)

A Choice of Futures

In 1995 there were three possible IP futures; let's call them heaven, hell, and purgatory. Heaven is where IPv6 displaces NAT and the Internet runs as it was meant to. Purgatory is where all the NAT problems are solved, but there is no demand for IPv6 as a result. Hell is where NAT problems are not solved, but IPv6 never takes off anyway. By choosing to discourage NAT, the IETF increased the chance of getting to heaven, but also created the risk of ending up in hell, which is what it's been so far.

The commercial world has not embraced IPv6, but NAT has taken over big time. In the latter half of the 1990s, the IETF standardized major protocols that did not work through NAT — notably, mobile IP,[5] IPsec,[6] and the session initiation protocol (SIP).[7] Far from discouraging NAT, this intentional disregard (defiance?) has, if anything, hurt these protocols.

Because it allows outgoing connectivity — a client in a private network can connect to a server in the global Internet — NAT has probably done very little to hurt the client-server protocols that are the Internet's bread and butter. As mentioned, however, NAT makes it difficult or impossible to deploy P2P protocols. Thus, it is hard to assess the damage NAT (or the IETF's refusal to accommodate it) has inflicted on the Internet.

Living with NAT

Client-server protocols have been enough to fuel Internet growth, and so the commercial world has been satisfied with IPv4. If there were actually a danger of running out of IPv4 addresses, the threat of this catastrophe might be enough to drive IPv6 deployment, but this just isn't going to happen. NAT provides enough addresses for each human on the planet to have an endless number of private connections and about 250 simultaneous active "global" connections. (This number is derived using the method in RFC 3194, assuming a 47-bit address space — 32 bits of IP, 15 bits of port.) IP addresses are more like radio spectrum than crude oil: rather than someday running out, we'll just get better at managing and reusing them.

Unlike the biblical hell, it is possible to crawl out of IP hell, and fortunately, this is finally happening. Standards are now in place or under development to define how to run mobile IP and IPsec through NATs, by tunneling over the user datagram protocol (UDP). More notably, the IETF's architecture oversight committee, the IAB, is permitting work on standard solutions to NAT's incoming-connectivity problem. Like a parent permitting a daughter's marriage to prevent her from eloping, however, the IAB isn't exactly blessing this work. In a nice bit of acronymship, these solutions must declare themselves as UNSAF (unilateral self-address fixing),[7] by limiting their own scopes and describing their own exit strategies.

Two projects are working to solve the incoming-connectivity problem — one in the context of SIP, and the other, ironically, in the context of IPv6. Both are based on Dean Kegel's groundbreaking work from 1998 at the Internet gaming company Activision. Like many breakthrough technologies, the idea is simple: a game client behind a NAT box (say, Bob's) contacts the game server, which records the global address and port the NAT box assigned to Bob. When Steve later wants to play a game with Bob, the game server tells Steve's game client which address and port to use to talk to Bob's game client, and the client makes the connection.

The commercial world has not embraced IPv6, but NAT has taken over big time.

SIP Solution

SIP is the signaling protocol of choice for establishing voice-over-IP (VoIP) calls. And it is fast becoming the protocol of choice for open presence and messaging.

The SIP-based NAT solution is the more interesting of the two because, I believe, it constitutes a legitimate next-generation architecture. SIP has a long-term stable identifier, called the SIP uniform resource identifier (URI), which functions like the IP address did originally. The SIP URI looks, not coincidentally, like an email address (for example, francis@cs.cornell.edu). In fact, it is substantially more powerful than the IP address because it can identify a user or a process rather than just a machine. Another person or computer can use your SIP URI to find you on multiple devices and follow you when you move (your desktop, your phone, your PDA, or a different desktop later). IP can't do that.

The SIP-based NAT solution consists of a protocol called simple traversal of UDP through NATs (STUN), which allows a SIP host to learn its own NAT-assigned address and port, and a protocol called interactive connectivity establishment (ICE), which describes how SIP uses STUN to establish peer connections. (See www.ietf.org/html.charters/sip-charter.html for pointers to these works.)

These protocols come mainly from Jonathan Rosenberg, chief scientist at Dynamicsoft, a leading SIP vendor. He is a coauthor of SIP and, arguably, the person most responsible for its success. Rosenberg is not pushing these protocols as a complete next-generation architecture, and I'm not sure he views them as such.

As a legitimate, cohesive, and complete architecture, however, the SIP-based NAT solution should have a name, and I like to call it NUTSS, after its core component pieces:

- *NAT* effectively extends the IP address space.
- *URIs* restore end-to-end stable addressing.
- *Tunnels* allow protocols like IPsec and mobile IP (and even TCP) to run through NATs over UDP.
- *SIP* routes messages with URIs, end-to-end, and lets hosts signal their intentions to each other in real time.
- *STUN* tells hosts how to establish direct IP connectivity through NATs.

Recognizing NUTSS as a cohesive architecture and naming it as such is important because IPv6 is unlikely to happen, and the Internet community needs to think about alternatives. If we think of STUN and ICE narrowly as nothing more than a way to make SIP work better, then only SIP applications will be able to take advantage of these advances. The network APIs to operating systems won't have socket equivalents that allow any network application to use NUTSS. More importantly, STUN and ICE work only for UDP, not TCP. The broader view that NUTSS provides would let us define TCP over UDP and get it into OS network stacks.

IPv6 Solution

The IPv6-based NAT solution is called Teredo. It lets an IPv6 host connected to a private IPv4 network obtain an IPv6 address. Teredo sends and receives IPv6 packets through NAT by tunneling them over UDP and IPv4. I do not believe Teredo is as legitimate a next-generation solution as NUTSS because it does not provide a long-term stable address. It produces transient IPv6 addresses that embed the address and port dynamically assigned by the NAT box. Teredo implicitly relies on some other naming space, such as DNS, for long-term identifiers. It is, therefore, an incomplete solution.

On the other hand, because it rides on the back of IPv6, Teredo does provide a general network API (that is, the IPv6-sockets API). This, for instance, lets TCP work over Teredo (TCP-IPv6-UDP-IP). Microsoft is using Teredo in its P2P application toolkit for XP, and has built a P2P messaging and communications application over it, called Three Degrees (www.threedegrees.com). However, it is still unlikely that anyone developing a client-server protocol would use the Teredo stack.

From a deployment perspective, the main impediment might be that firewalls won't allow the packets through because they don't recognize the stack. Of course, Teredo's success is not the same as IPv6's success, because it can operate without a single IPv6 router. The IPv6 founders did not develop IPv6 to be a glorified NAT solution.

Conclusions

The Internet community is on the cusp of an important change, and it will be interesting to see how it plays out. I believe the best outcome would be for the entire community to put its weight behind either Teredo or NUTSS. This would create the necessary synergy to fill in all the architecture's missing pieces, such as full API support in the OS, and firewalls that recognize the appropriate stacks and protocols. I prefer NUTSS, but will take Teredo.

The worst, and perhaps more likely, future is that Microsoft supports Teredo in XP, but gets only a small following. Traditional SIP applications use STUN, but nothing else does. Most P2P applications continue to use ad hoc and proprietary NAT tunneling hacks.

To avoid this latter scenario, the IETF leadership needs to recognize that there is a good chance that IPv6 will never happen. (After all, if the market didn't pick it up when the NAT problem was unsolved, why would it move to it once solutions are in place?) The IAB should no longer be satisfied with simply labeling NAT solutions as UNSAF, because the commercial world just doesn't care. Rather, it needs to start hedging its bets, and take a serious look at one of these NAT-based approaches as a valid next-generation architecture.

If the Internet isn't going to heaven, let's at least hope it goes NUTSS.

References

1. V. Fuller et al., "Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy," Internet Eng. Task Force RFC 1519, Sept. 1993; www.rfc-editor.org/rfc/rfc1519.txt.
2. Y. Rekhter et al., "Address Allocation for Private Internets," Internet Eng. Task Force RFC 1918, Feb. 1996; www.rfc-editor.org/rfc/rfc1918.txt.
3. K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," Internet Eng. Task Force RFC 1631, May 1994; www.rfc-editor.org/rfc/rfc1631.txt.
4. P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," Internet Eng. Task Force RFC 3022, Jan. 2001; www.rfc-editor.org/rfc/rfc3022.txt.
5. C. Perkins, "IP Mobility Support," Internet Eng. Task Force RFC 2002, Oct. 1996; www.rfc-editor.org/rfc/rfc2002.txt.
6. S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," Internet Eng. Task Force RFC 2401, Nov. 1998; www.rfc-editor.org/rfc/rfc2401.txt.
7. J. Rosenberg et al., "SIP: Session Initiation Protocol," Internet Eng. Task Force RFC 3261, June 2002; www.rfc-editor.org/rfc/rfc3261.txt.
8. L. Daigle, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) across Network Address Translation," Internet Eng. Task Force RFC 3424, Nov. 2002; www.rfc-editor.org/rfc/rfc3424.txt.

Paul Francis is an associate professor of computer science at Cornell University. His research interests include overlay networks and Internet routing and addressing. He received a PhD in computer science from University College, London. Francis has done research at MITRE, Bellcore, NTT Software labs in Tokyo, and the AT&T Center for Internet Research at ICSI (ACIRI). Contact him at francis@cs.cornell.edu.