The ISP Column
A monthly column on things Internet

April 2009

Geoff Huston

NAT++: Address Sharing in IPv4

In this article I'd like to look at the topic that was presented at a session at the 74th meeting of the IETF in March 2009, on Address sharing (the SHARA BOF), and look at the evolution of NAT architectures in the face of the forthcoming depletion of the unallocated IPv4 address pool.

> This particular story about hybrid NATs has is its moments of delicious irony. For many years I sat through presentations in the IETF, and then in IPv6 meetings that loudly and proudly proclaimed the evil of Network Address Translation. The anti-NAT hysteria reached its zenith for me in 2004 when I heard in an IPv6 event that NATs evidently caused human fatalities. Disregarding these hysterical outbursts, it is certainly the case that the widespread use of such pejorative expressions to describe NATs and the long-delayed standardization of NAT behaviour over the past decade has been at best unhelpful to this industry. At present, when we are facing a network that is critically reliant on NATs in order to keep working across a looming period of address scarcity, it looks like NATs are now a critical necessity. So it seems that the IPv6 sales message that we needed an end-to-end IPv6 network in order to avoid a NATted Internet and avoid all those nasty NAT behaviours has now been transformed into a message that we need NATs just as much, if not more, than we need IPv6 right now. How ironic.

Within the next couple of years we will have run out of the current supply of IPv4 addresses. As of the time of writing this column the projected data when IANA is depleted is 27th June 2011, and the first RIR to deplete its address pool will occur on the 21st March 2012

> A couple of years ago the increasing consumption of IPv4 addresses implied that the projected exhaustion date of the unallocated IPv4 address pool was getting closer in time, but the effects of the current global financial stuff up have had an impact on address consumption rates, and recently the exhaustion date has been moving further into the future once more, in line with the slower pace of IPv4 network expansion.
>
> If may not be ironic, but it would certainly be an interesting coincidence if exhaustion of the IPv4 address space occurred on the 21st of December 2012 if only to claim that the Mayan Calender had got it right after all! (http://en.wikipedia.org/wiki/Mayan_calendar)

Irrespective of the precise data of depletion, the current prediction is that the consumption of addresses at that time when then free pool of addresses is exhausted will probably be running at some 220 million addresses per year, indicating a deployment rate of some 170 - 200 million new services per year using IPv4. The implication is that the Internet will hit the brick wall of exhaustion while operating its growth engines at full speed.

How quickly will IPv6 come to the rescue? Even the most optimistic forecast of IPv6 uptake for the global Internet is measured on years rather than months following exhaustion and the more pessimistic forecasts extend into multiple decades.

> For one such analysis using mathematical modelling techniques, see Jean Camp's work at http://www.ripe.net/ripe/meetings/ripe-56/presentations/Camp-IPv6_Economics_Security.pdf. One of the conclusions from that 2008 study is: "There is no feasible path which results is less than years of IPv4/IPv6 co-existence. Decades is not unreasonable."

The implication of this is that we will need to operate a dual stack Internet for many years yet, and the associated implication is that that we are going to have to make the existing IPv4 Internet span a billion or more new deployed services, and do so without any additional address space.

So how are we going to make the IPv4 address pool stretch across an ever-larger Internet?

Given that the tool chest we have today is the only one available, then it looks like there is only one answer to this question: Network Address Translation, or NATs.

> For a description of how NATs work and some of the terminology used to describe NAT behaviour, see "Anatomy: A Look inside Network Address Translators" <http://www.potaroo.net/papers/ipj/2004-v7-n3-nat/nats.html">

Today NATs are predominately edge devices that are bundled with DSL modems for residential access, or bundled with routing and security firewall equipment for small to medium enterprise use as an edge device. The generic model of NAT deployment these days is a small scale edge device that generally has a single external-side public IP address and an internal-side private IP network address (often network 10), and the NAT performs address and port translation to map all currently active sessions from the internal addresses to ports on the public IP address. This NAT deployment assumes that each edge customer has the unique use of a public IP address.

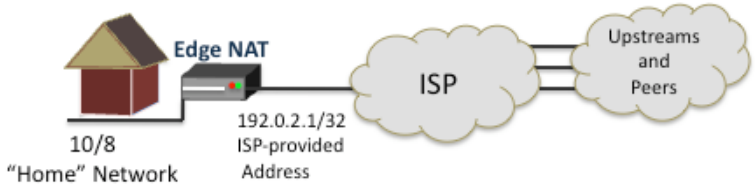

Figure 1 - Conventional NAT deployment

The question provoked by IPv4 address exhaustion is what happens when there are no longer sufficient IPv4 addresses to provide this 1:1 mapping between customers and public IPv4 addresses? In other words what happens when there are simply not enough IPv4 addresses to allow every customer have exclusive use of their own unique IPv4 address?

There are really only two possible answers to this question. One is for noone to use IPv4 addresses at all, on the basis that the entire Internet has migrated to use IPv6. But this answer appears to be an uncomfortable number of decades away, so we need to examine the other answer. And that answer is that if there are not enough addresses to go around then we are going to have to share them.

Isn't this impossible in the Internet architecture? The IP address in the header determines the packet's destination. If two or more endpoints are sharing the same address then how is the network going to figure out which packets go to which endpoint? Its here that NATs and the transport layer protocols TCP and UDP come together. The approach is to use the port address in the TCP and UDP header as the distinguishing element.

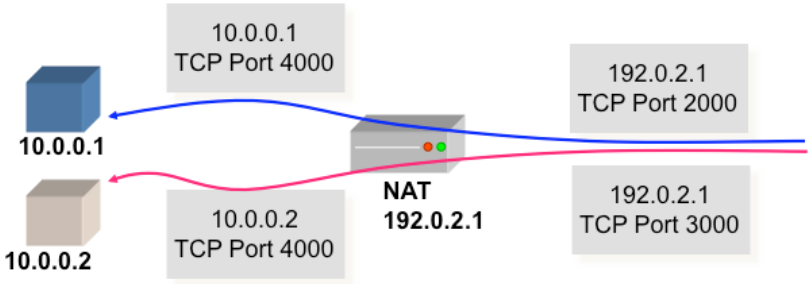

Figure 2 - Address Sharing with NATS

For example, incoming TCP packets with TCP port address 2000 may need to be directed to endpoint A, while incoming TCP packets with TCP port address 3000 need to be directed to endpoint B. The incoming TCP packets with a port address of 2000 are translated to have endpoint A's private IP address, while incoming TCP packets with a port address of 3000 will be translated to have endpoint B's private address. As long as you restrict yourself to applications that use TCP or UDP, and as long as you don't rely on receiving ICMP packets, and as long as you don't use applications that contain IP addresses in their payload, then you might expect the application to function.

> ICMP is a problem because the ICMP packet does not contain a TCP or UDP transport layer. All that a NAT will see in the ICMP packet is its own external address as the destination IP address. To successfully deliver an ICMP packet through a NAT, the NAT needs to perform a more complex function that uses the ICMP-encapsulated IP header to select the original outbound combined IP + TCP header or IP + UDP header in the ICMP payload. The source IP address and transport protocol port address in the ICMP payload are then used to perform a lookup into the NAT binding table and the perform two mappings: one on the ICMP header to map the destination IP address to the internal IP address, and the second on the payload header where the source IP address and port number are changed to the interior-side values, and the checksums altered as appropriate. Now in most cases ICMP really is not critical, but one message that is important is the ICMP packet-too-large-and-fragmentation-disabled message used in IPv4 path MTU discovery.

That's fine in theory, but how can this be achieved in practice? How can many customers, already using NATs, share a single public IP address?

Carrier Grade NATs

One possible response is to add a further NAT into the path. In theory the Service Provider (SP) could add NATs on all upstream and peer connections, and perform an additional NAT operation as traffic enters and leaves the SP's network. Variations of this approach are possible, placing the SP NATs at customer aggregation points within the SP's network, but the principle of operation of the SP NAT is much the same.

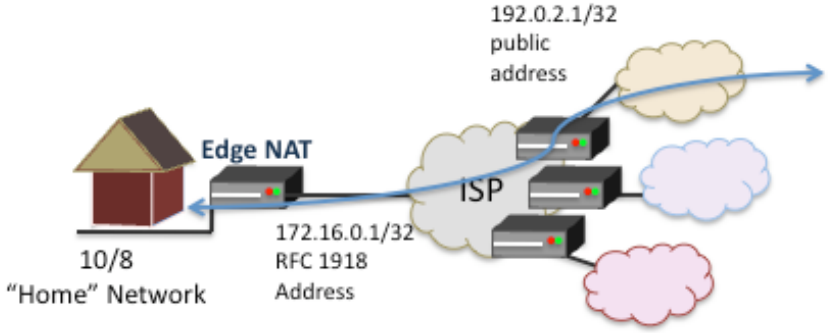

Figure 3 - Carrier NATs

The edge NATs translate between private address pools at each customer's site and an external address provided by the Service Provider (SP), so nothing has changed there. The change in this model is that the SP places a further NAT in the path within the SP network, so that a set of customers are then sitting behind a larger NAT insider the SP's network.

This implies that the external address provided by the SP to the customer is actually yet another private address, and the SP's NAT performs yet another transform to a public address in this second NAT. In theory at any rate this is just a larger version of an existing NAT with larger NAT binding space, higher packet processing throughputs, and a comprehensive specification of NAT binding behaviour. In practice this may be a little more complicated in that at the edge of the network the packet rates are well within the processing capability of commodity processors, while in the core of the network there is an expectation of higher levels of robust performance from such units. As it is intended that such NATs handle thousands of customers and large numbers of data flows and peak packet rates, so it has been termed a "*Carrier Grade NAT*" (CGN), or "*Large Scale NAT*" (LSN).

From the inside not much has changed with the addition of the CGN in terms of application behaviour. It still requires an outbound packet to trigger a binding that would allow a return packet through to the internal destination, so nothing has changed there. Other aspects of NAT behaviour, notably the NAT binding lifetime and the form of "*cone behaviour*" for UDP take on the more the more restrictive of the two NATs in sequence. The binding times are potentially problematical in that the two NATs are not synchronised in terms of binding behaviour. If the CGN has a shorter binding time, it is possible for the CGN to misdirect packets and cause application level hang ups. However this is not overly different to a single level NAT environment where aggressively short NAT binding times will also run the risk of causing application level hang ups when the NAT drops the binding for a active session that has been quiet for an extended period of time.

However, one major assumption is broken in this structure, namely that an IP address is associated with a single customer. In this model a single public IP address may be simultaneously used by many customers at once, albeit on different port numbers. This has obvious implications in terms of some current practices in filters, firewalls, "black" and "white" lists and some forms of application level security and credentials where the application makes an inference about the identity and associate level of trust in the remote party based on the remote party's IP address.

This approach is not without its potential operational problems as well. For the SP service resiliency becomes a critical issue in so far moving traffic from one NAT-connected external service to another will cause all the current sessions to be dropped, unless the internal SP network architecture uses a transit access network between the CGNs and the external transit providers. Another issue is one of resource management in the face of potentially hostile applications. For example, an end host infected with a virus may generate a large amount of probe packets to a large range of addresses. In the case of a single edge NAT the large volumes of bindings generated by this behaviour become a local resource management problem as the customer's network is the only impacted site. IN the case where an CGN is deployed the same behaviour will start to consume binding space on the CGN and, potentially, can starve the CGN of external address bindings. If this problem is seen to be significant the CGN would need to have some form of external address rationing per internal client in order to ensure that the entire external address pool is not consumed by a single errant customer application.

The other issue here is one of scalability. While the greatest leverage of the CGN in terms of efficiency of utilisation of external addresses occurs when the greatest numbers of internal edge NATted clients are connected, there are some real limitations in terms of NAT performance and address availability when a SP wants to apply this approach to networks where the customer population is in the millions or larger. In this case the SP is required to use an IPv4 private address pool to number every client. But if network 10 is already used by each customer as their "internal" network, then what address pool can be used for the SP's private address space? One of the few answers that come to mind is to deliberately partition the network into a number of discrete networks, each of which can be privately numbered from 172.16.0.0/12, allowing for some 600,000 or so customers per network partition, and then use a transit network to "glue" together the partitioned elements.

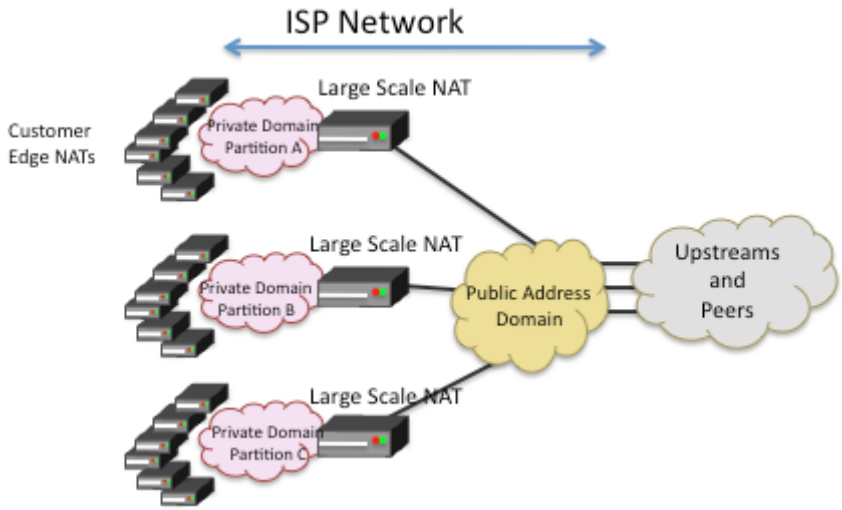

Figure 4 - Multiple Carrier NAT Deployment using Network Partitioning

The advantage of the CGN approach is that for the customer nothing changes. There is no need for any customers to upgrade their NAT equipment or change them in any way, and for many service providers this is probably sufficient motivation to head down this path. The disadvantages of this approach lie in the scaling properties when looking at very large deployments, and the issues of application-level translation, where the NAT attempts to be "helpful" by performing deep packet inspection and rewriting what it thinks are IP addresses found in packet payloads. Having one NAT do this is bad enough, but loading them up in sequence is a recipe for trouble!

Are there alternatives?

Dual-Stack List and Carrier Grade NATs

One rather elegant alternative is described by Alain Durand and others in an Internet Draft "Dual-stack lite broadband deployments post IPv4 exhaustion" (draft-ietf-softwire-dual-stack-lite-00.txt). The assumption behind this approach is that the SP's network infrastructure will need to support IPv6 running in native mode in any case, so is there a way in which the SP can continue to support IPv4 customers without running IPv4 internally?

Here the customer NAT is effectively replaced by a tunnel ingress/egress function in the Dual-stack lite home gateway. Outgoing IPv4 packets are not translated, but are encapsulated in an IPv6 packet header, where the IPv6 packet header contains a source address of the carrier side of the home gateway unit, and a destination address of the ISP's Gateway unit. From the Service Provider's perspective each customer is no longer uniquely addressed with an IPv4 address, but instead is addressed with a unique IPV6 address, and provided with the IPv6 address of the provider's combined IPv6 tunnel egress point and IPv4 NAT unit.

The Service Provider's Dual-Stack Lite gateway unit will perform the IPv6 tunnel termination and a NAT translation using an extended local binding table. The NAT's "interior" address is now a 4-tuple of the IPv4 source address, protocol ID, and port, plus the IPv6 address of the home gateway unit, while the external address remains the triplet of the public IPv4 address, protocol ID and port. In this way the NAT binding table contains a mapping between interior "addresses" that consist of IPv4 address and port plus a tunnel identifier, and public IPv4 exterior addresses. This way the NAT can handle a multitude of net 10 addresses, as they can be distinguished by different tunnel identifiers. The resultant output packet following the stripping of the IPv6 encapsulation and the application of the NAT function is an IPv4 packet with public source and destination addresses. Incoming IPv4 packets are similarly transformed, where the IPv4 packet header is used to perform a lookup in the D-S Lite Gateway unit, and the resultant 4-tuple will be used to create the NAT-translated IPv4 packet header plus the destination address of the IPv6 encapsulation header.

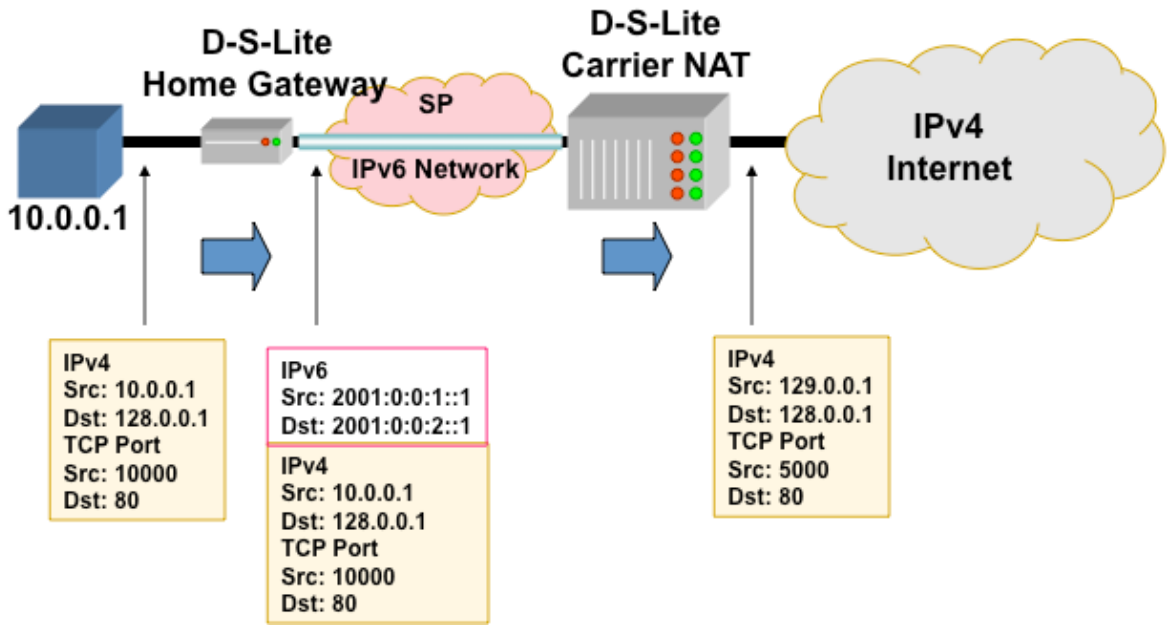

Figure 5 Dual-Stack Lite

The advantage of this approach is that there now only needs to be a single NAT in the end-to-end path, as the functionality of the customer NAT is now subsumed by the carrier NAT. This has some advantages in terms of those messy "value-added" NAT functions that attempt to perform deep packet inspection and rewrite IP addresses found in data payloads. There is also no need to provide each customer with a unique IPv4 address, public or private, so that the scaling limitations of the dual-NAT approach are also eliminated. The disadvantages of this approach lie in the need to use a different CPE device, or at least one that is reprogrammed. The device now requires an external IPv6 interface and at the minimum a IPv4 / IPv6 tunnel gateway function. The device can also include a NAT it so desired, but this is not required in terms of the basic dual-stack lite architecture.

This approach pushes the translation into the middle of the network, where the greatest benefit can be derived from port multiplexing, but it also creates a critical hotspot for the service itself. If the carrier NAT fails in any way then the entire customer base is disrupted. It seems somewhat counter-intuitive to create a resilient network with stateless switching environments and then place a critical stateful unit in the middle! So is there an approach that can push this translation back to the edges while avoiding a second NAT in the carrier's network?

The Address plus Port Approach

The observation here is that CPE NATs currently map connections into the 16 bit port field of the single external address. If the CPE NAT could be coerced into performing this mapping into 15 bits of the port field, then the external address could be shared between twp edge CPEs, with the leading bit of the port field denoting which CPE. Obviously, moving the bit marker across the port field will allow more CPEs to share the one address, but reduce the number of available ports for each CPE in the process.

The theory is again quite simple. The CPE NAT is dynamically configured with an external address, as happens today, and a port range, which is the additional constraint. The CPE NAT performs the same function as before, but it is now limited in terms of the external ports it can use in its NAT bindings to those that lie within the provided port range, as some other CPE may be concurrently using the same external IP address with a different port range.

For outgoing packets this implies only a minor change to the network architecture, in that the Radius exchange to configure the CPE now must also provide a port range to the CPE device. However the case of incoming packets is more challenging. Here the SP must forward the packet based not only on the destination IP address, but also on the port value in the TCP or UDP header. A convenient way to do this is to take the dual-stack lite approach and use a IPv4 in IPv6 tunnel between the CPE and the external gateway. This gateway, or "A+P Router" needs to be able to associate each address and port range with the IPv6 address of a CPE, which it can learn dynamically as it decapsulates outgoing packets. Corresponding incoming packets are encapsulated in

IPv6 using the IPv6 destination address that it has learned previously. In this manner the NAT function is performed at the edge, much as it is today, and the interior device is a more conventional form of tunnel server.

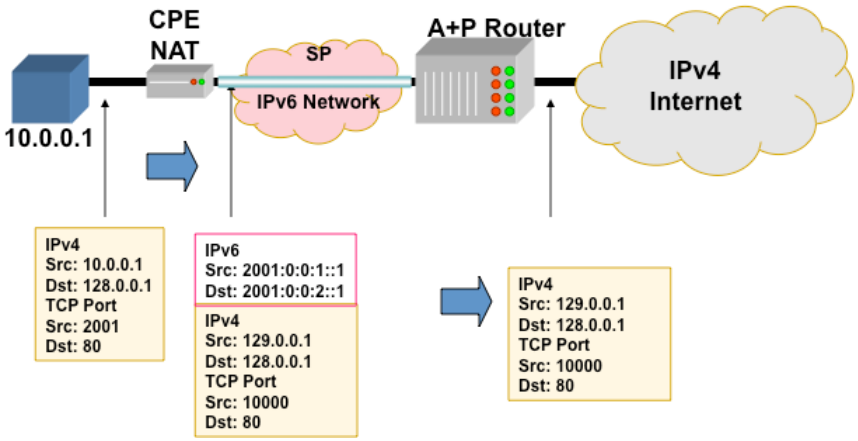

Figure 6 - Address Plus Port Framework

This approach relies on every CPE device being able to operate using a restricted port range, and able to perform IPv4-in-IPv6 tunnel ingress / egress functions, and act as an IPv6 provisioned endpoint for the SP network, which is perhaps an unrealistic hope. Further modifications to this model propose the use of an accompanying CGN operated by the SP to handle those CPE devices that cannot support this address plus port functionality.

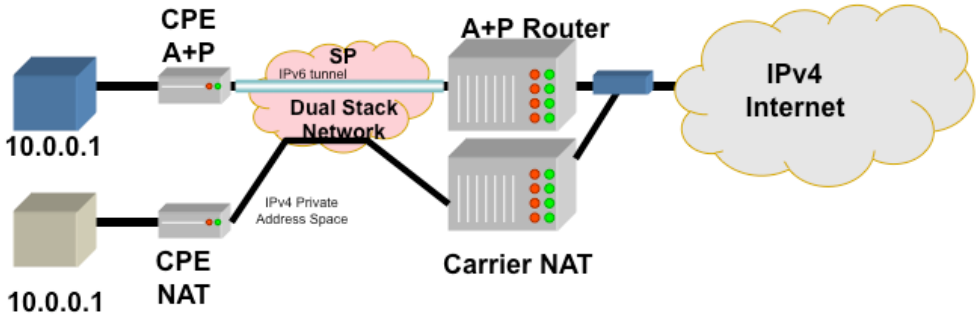

Figure 7 - Combined Address Plus Port and CGN

If the port range assigned to the CPE is from a contiguous range of port values then this approach could exacerbate some known problems with infrastructure protocols. The DNS problems with guessable responses (The so-called "Kaminsky attack" on the DNS <http://www.doxpara.com/DMK_BO2K8.ppt> and <http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html>) is one such example where the attack can be deflected, to some extent, by using a randomly selected port number for each DNS query. Restricting the port range could mitigate the efficacy of such measures under certain conditions.

However, despite such concerns, the approach has some positive aspects. Pushing the NAT function to the edge has some considerable advantage over the approach of moving the NAT to the interior of the network. The packet rates are lower at the edge, allowing for commodity computing to process the NAT functions across the offered packet load without undue stress. The ability to control the NAT behaviour with the Internet Gateway Device protocol as part of the uPnP framework will still function in an environment of restricted port ranges. Aside from the initial provisioning process to equip the CPE NAT with a port range, the CPE, and the edge environment is largely the same as today's CPE NAT model.

That is not to say that this approach is without its negative aspects, and its unclear as to whether the perceived benefits of a "local" NAT function outweigh the problems in this particular model of address sharing. The concept of port "rationing" is a very suboptimal means of address sharing, given that once a CPE has been assigned a port range, those port addresses are unusable by any other CPE. The prudent SP would assign to each CPE a port address pool equal to some estimate of peak demand, so that, for example, each CPE would

be assigned 1,000 ports, allowing a single external IP address to be shared across only 60 such CPE clients. The Carrier Grade NAT approach or the Dual-Stack Lite approach does not attempt this form of rationed allocation, allowing the port address pool to be treated as a common resource, with far higher levels of utilization efficiency. The difference here parallels the difference in efficiency between circuits and packets at layer 2 in the network model. So the leverage obtained in terms of making efficient use of these additional 16 bits of address space is reduced by the imposition of a fixed boundary between customer and SP use. The central NAT model effectively pools the port address range and would for more efficient sharing of this common pool across a larger client base.

> Alain Durand reported to IETF 74 on a data collection experiment using a CMTS with 8,000 subscribers where the peak port consumption level was 40,000 ports, or a maximum average port consumption of 5 ports per subscriber in each direction. As Alain noted, this average value needs to be compared with the hundreds of ports consumed by a single client browsing a Web 2.0/AJAX site, but it does note that a central model of port sharing does yield far higher levels of address sharing efficiency than the address plus port advanced allocation model.
> (http://www.ietf.org/proceedings/09mar/slides/shara-8/shara-8.htm)

The other consideration here is that this approach is a higher overhead for the SP, in that the SP would have to support both 'conventional' CPE equipment and Address plus Port equipment. In other words the SP will have to deploy a CGN and support customer CPE using a two level NAT environment in addition to operating the Address plus Port infrastructure. Unless customers would be willing to pay a significant price premium for such address plus port service it is unlikely that this option would be attractive for the SP as an additional cost over and above the CGN cost.

General Considerations with Address Sharing

The basic elements of any such approach to address sharing involve the CPE equipment at the edge, optionally some form of tunnelling of traffic between the CPE and the carrier equipment, and carrier-provided equipment at the edge or the carrier's network.

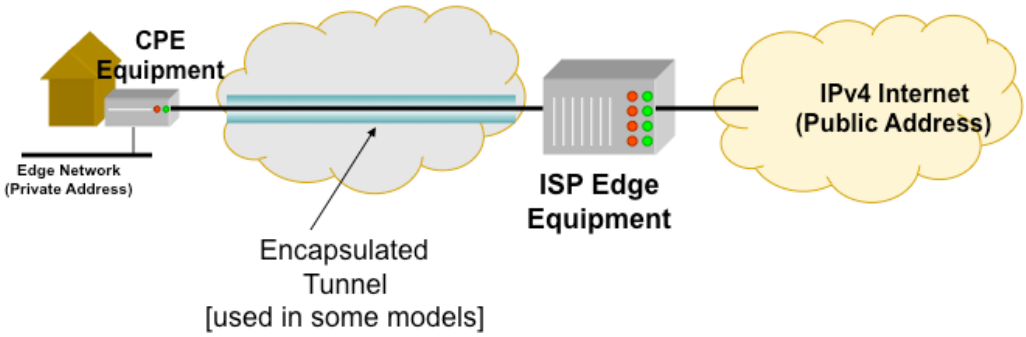

Figure 8 - Generic Architecture for Address Sharing

There are a variety of technical solutions here that involve these basic building blocks, so its not true to say that this is a technical challenge. So what is the problem here?

I suspect that the real issue here is the consideration of the relevant business model that would drive this deployment. Today's Internet is large. It encompasses some 1.7 billion human users, a larger pool of devices, and hundreds of millions of individual points of control. If we want to change this deployed system then we will need copious quantities of money, time and unity of purpose. So do we have money, time and unity of purpose?

Money is missing. It could be argued that we have left the entire IPv6 transition effort to this late stage in the day because of a lack of money. The Internet's major leverage was that it was cheap. Packet sharing is intrinsically more efficient than circuit sharing and the shift in functionality of network service management from the network to the customer-owned and operated end points implied further cost savings for the network operator. So the Internet model gained ascendency because for consumers it represented a cost effective choice. It was cheap. But what does IPv6 offer consumers? For existing Internet consumers it appears that IPv6 does not offer anything that they don't already have with IPv4 - it offers mail, the web, various forms of voice services and games. So consumers are not exactly motivated to pay more for the same services they enjoy today. So it would appear that the ISP has to carry this cost without incremental revenue from its customer base. But the ISP industry has managed to shave most of its revenue margins in a highly competitive industry, and at the same time lose control of services, their delivery, and their potentially lucrative revenue margins. So the ISP industry has been collectively sitting on its hands not because it cannot see the problem in terms of the imminent exhaustion of IPv4, but because it has little choice due to financial constraints that have prevented it from making the necessary longer term investments in IPv6. So if the ISP industry has been unwilling to invest in IPv6 so far, then what incentive is there for the ISP industry to invest in IPv6 and at the same time also invest in these IPv4 address sharing measures? Is the lure of new low margin customers sufficient incentive to make such investments in this carrier-grade equipment? Or is the business case still insufficiently attractive?

Time is missing. The unallocated IPv4 address pool is already visibly waning. Without any form of last minute rush the pool will be around for the next 3 years, or until 2012 or so. But with any form of typical last minute rush this pool could be depleted in the coming months rather than the coming years. Can we do what we need to do to get any of these approaches to a state of mass market deployment in the next few months? All these approaches appears to be at the early stages of a timeline that starts with research and then moves on to development and prototyping and trials, then to standards activity and industry engagement to orchestrate supply lines for end user equipment, ISP equipment and definition of operational practices and then to product and service development and, finally to deployment. For an industry that is the size of the Internet "technical agility" is now an obsolete historical term. Even with money and unity of purpose this will take some years, And without money, or even the lure of money, it becomes a far more protracted process, as we've seen already with IPv6 deployment.

And do we have unity of purpose here? Do we agree on an approach to address sharing that will allow each player to perform their tasks? To allow consumer product vendors to develop the appropriate product? To allow application developments to develop applications that will operate successfully in this environment? To allow the end user platform vendors to incorporate the appropriate functionality in the operating system stacks? To allow ISPs to integrate vendor's productions into their operational environments? Right now its pretty clear that what we have is a set of ideas, each of which has relative merits and disadvantages, and no real unity of purpose.

Its easy to be pessimistic at this stage, given that the real issues here appear to be related more to the issues associated with a very large industry attempting to respond to a very challenging change in the environment in which it operates. The issue here is not really whether Address Plus Port routing is technically inferior to Dual-Stack Lite, or whether Carrier Grade NATs are technically better or worse than either of these approaches. The issue here is whether this industry as a whole is going to be capable of sustaining its momentum and growth across this hiatus. And, from this perspective, I believe that such pessimism about the future of the Internet is unwarranted. The communications industry has undergone significant technological changes over the years, and this is one more in the sequence. Some of these transformations have been radical in their impact, such as the introduction of the telephone in the late nineteenth century, while others have been more subtle, such as in the introduction of digital technology to telephony in the latter part of twentieth century replacing the earlier analogue circuit model of telephony carriage. Some changes have been associated with high levels of risk and we've seen a myriad of smaller more agile players enter the market to lead the change while the more risk averse enterprises stand back, while other changes require the leverage of economies of scale and we've seen market consolidation behind a smaller number of highly capitalised players.

I must admit that I'm personally quite taken with the Dual-Stack Lite approach, but I suspect that the lowest common denominator fall back position that this somewhat conservative industry will adopt is one that will rely strongly on Carrier Grade NATs and the industry will, on the whole, eschew the more complex support mechanisms required by the various permutations of address plus port routing. But that's just my opinion. What's yours?

Further Reading

The Address Sharing BOF was held at IETF 74 in March 2009. The presentations, and a summary of the session can be found as part of the proceedings of that meeting:
http://www.ietf.org/proceedings/09mar/shara.html

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre, nor those of the Internet Society.

About the Author

GEOFF HUSTON is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. he graduated from the Australian National University with a B.Sc, and M.Sc. in Computer Science. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001.

http://www.potaroo.net