The ISP Column
A monthly column on things Internet

March 2011
Geoff Huston

Transitioning Technologies – Part 2

This month we are continuing to look behind the various opinions and perspectives about the transition to IPv6, and look in a little more detail at the nature of the technologies being proposed to support the transition to IPv6.

After some time of hearing dire warnings about the imminent exhaustion of the stocks of available IPv4 address space, we've now achieved the first milestone of address exhaustion, the depletion of the central pool of IANA-managed address space. The last 5 /8s were handed out from IANA to the RIRs on the 3rd February. After some years of industry-wide general inattention and inaction with IPv6 perhaps it's not unexpected to now see a panicked response along the lines of "Maybe we should do something now!"

But what exactly should be done? It's one thing to decide to "support" IPv6 in a network, but quite another to develop a specific plan, complete with specific technologies, timelines, costs, vendors, and a realistic assessment of the incremental risks and opportunities. While working through some of this detail has the normal levels of uncertainty that you would expect to see in any environment that is undergoing constant change and evolution, an additional level of uncertainty here is a by-product of the technology itself.

There's not just one approach to adding support for IPv6 in your network, but many. And it's not just one major objective you need to address, namely incremental deployment of IPv6 as second protocol into your operational network without causing undue disruption to existing services, but two, as the second challenging objective is how to fuel continued growth in your network service platform when the current supply lines of readily available IPv4 addresses effectively dry up.

The good news is that many folk have been busy thinking about these inter-twined objectives of extending the useful lifetime of IPv4 in the Internet and simultaneously undertaking the IPv6 transition, and there are a wealth of possible measures you can take, and a broad collection of technologies you can use. Fortunately, we are indeed spoilt with choices here!

The not so good news is that many folk have been busy thinking about these inter-twined objectives, and there are a wealth of possible measures you can take, and a broad collection of technologies you cans use. These options may, or may not, be optimal for your particular circumstances, and may, or may not, be useful for you in mitigating address depletion and may, or may not, be consistent with your chosen longer term network objectives. Unfortunately, we are spoiled for choices here!

Let's have a look at each of the major transitional technologies that are currently in vogue, and look at their respective strengths and weaknesses and their intended area of applicability. In the previous column we looked at this from the perspective of the end user. In this second next part of the article we'll look at this from the other side, looking at options for ISPs.

V6 for ISPs

While the "self-help" auto-tunnelling approaches for clients outlined in Part 1 of this article are a possible answer, their utility is appropriately restricted to a very small number of end clients who have the necessary technical expertise and who are willing to debug some rather strange resultant potential problems relating to asymmetric paths, third party relays, potential MTU mismatches and interactions with filters, it is not a reasonable approach for the larger Internet.

From the perspective of the mass market for Internet Services, we cannot assume that clients have the motivation, expertise and wherewithal to bypass their ISP and set up IPv6 access on their own, either through auto-tunnelling or through manually configured tunnels. The inference from this observation is that for as long as the mass market ISPs do not commit to IPv6 services, and for as long as they continue to stall in deploying services supporting dual access for their clients, then the entire IPv6 transition story remains effectively stalled.

How can ISPs support IPv6 access for their clients?

The Dual-Stack Service Network

Perhaps its blindingly obvious, but the most direct response here is for the ISP to operate a dual stack network.

And the most direct way to achieve this is for the ISP's infrastructure to support both IPv6 wherever there is IPv4, so that the delivery of services to the ISP's clients in IPv6 faithfully replicates the service offered in IPv4. This implies that the network needs to support IPv6 in the ISP's routing infrastructure, in the network's data plane, in the load management systems, in the operational support infrastructure, in access and accounting, and in peering and in transit. In short, wherever there is IPv4 there needs to be IPv6.

Drilling down just one level, the list of infrastructure elements that require dual stack service includes the routing and switching elements, including the internal and external routing protocols. The task includes negotiating peering and transit services in IPv6 to complement those in IPv4. Network infrastructure also includes VPN support and other forms of tunnels, as well as data centre front end units including load balancers, filters and firewalls, and various virtualised forms of service provision. The task also includes integration of IPv6 in the network management subsystem and the related network measurement and reporting system. Even a comprehensive audit of the supported MIBs in the network's active elements to ensure that the relevant IPv6 MIBs are supported is an essential task. A similar task is associated with equipping the server infrastructure with IPv6 support, and at the higher levels of the protocol stack are the various applications, including Web services, Mail, DNS, Authentication and Accounting, VOIP servers, load balancers, cloud servers and similar.

And that's just the common elements of most ISP's infrastructure. Every ISP also has more specialised elements in its service portfolio, and each one of these also requires a comprehensive audit to ensure that there is a IPv6 story for each and every one of these elements that leads to a comprehensive dual stack outcome.

As obvious as this approach might appear, there are two significant problems with this approach. Firstly, it requires a comprehensive overhaul of every element in the ISP's service network. Even for small scale ISPs this is not trivial, and for larger service provider platforms this is an exercise that may take months if not years and make considerable inroads into the operating budgets of the ISP. Secondly, it still does not take into account the inevitable fact that in the coming months the current supply lines of IPv4 addresses will come to a halt and any continued expansion of the service platform will require some different approaches to the way in which IPv4 addresses are deployed in the service platform.

While the approach of simply provisioning IPv6 alongside IPv4 in a simple dual-protocol service infrastructure may appear to be the most obvious response to the need to transition to IPv6, but it may not necessarily be the most appropriate response for many ISPs to the dual factors of IPv6 transition and IPv4 address exhaustion.

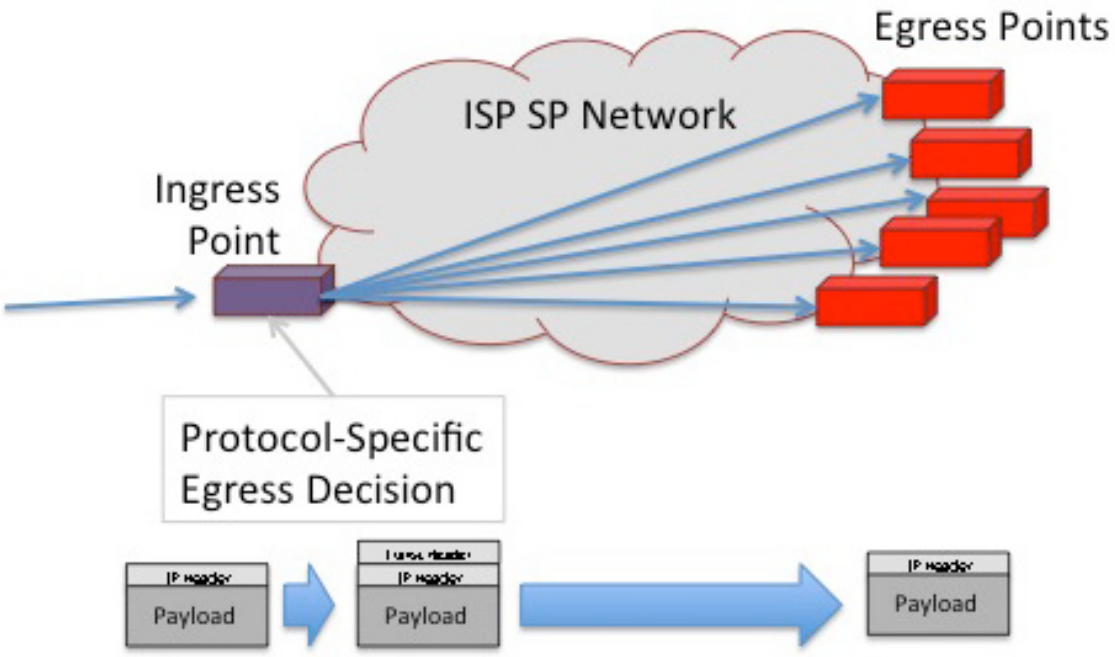
Are there alternative approaches for ISPs? Of course!

Hybrid Approaches

Saying that an ISP must deploy IPv6 across all of its infrastructure and actually doing it are often quite different. The cost of converting all parts of an ISP's operation to run in dual stack mode can be quite high, and the benefit of running every aspect of an ISP's service offering in dual stack mode is dubious at best.

Are there middle positions here? Is it possible for an ISP to deliver robust IPv6 services to clients will still operating an IPv4-only internal network?

One way to look at an ISP's network is as a transit conduit.



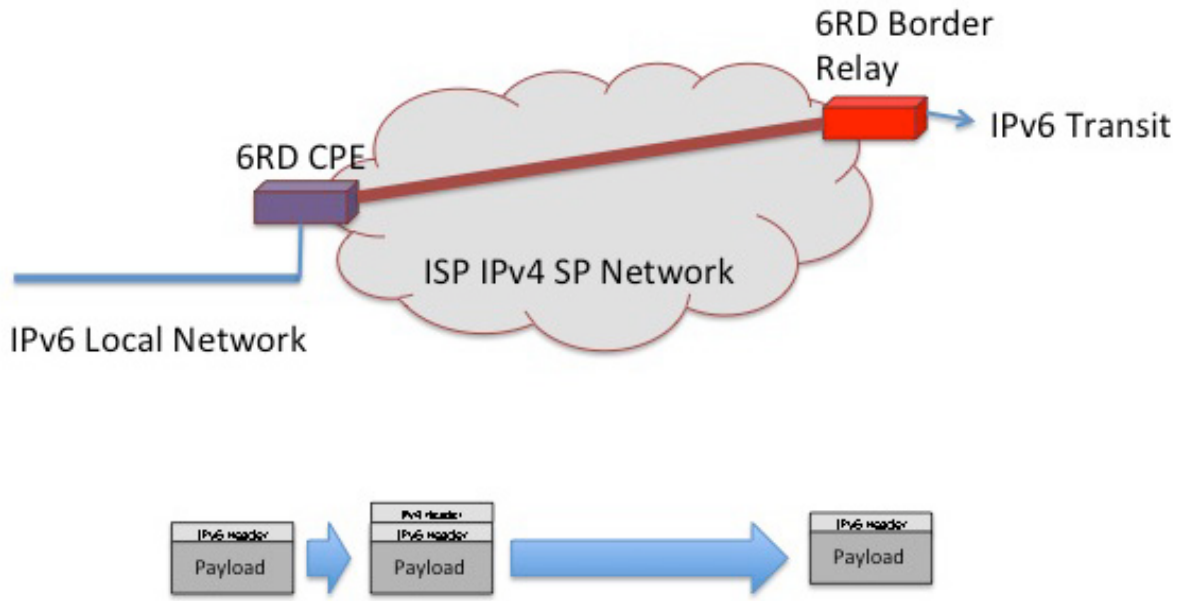
The ISP needs to be able accept packets from an external interface, determine the appropriate egress point for the packet within the context of the local network, and then ensure that the packet is passed out this egress interface. The internal network need not operate in the same protocol context as the protocol of the packets the network is handling. Viewed at a level of the minimal essentials, the network needs to be able to have some protocol-specific capability at its ingress points in order to determine each incoming packet's appropriate egress point, but thereafter during the transit of the SP network, the minimum necessary association to maintain is the identity of this pre-selected egress point with the packet. Now if the network uniformly supports the same protocol as the packet, then the same egress decision can be made at each forwarding point within the network. Alternatively, the packet can be encapsulated with an outer wrapper that identifies the egress point using the same protocol context as that used by the SP's internal switching elements, and the packet can be passed through the SP transit network using only this temporary wrapper to determine the sequence

of forwarding decisions. MPLS networks are an excellent example of this form of approach, as are other forms of IP-in-IP encapsulation. The advantage of this approach is that the service provider network's internal infrastructure need not be altered to support additional carriage protocols: the changes to specifically support IPv6 are required only at the network's ingress elements, and a basic encapsulation stripping function is used at all egress points.

With this in mind, lets have a look at some of these hybrid approaches to supporting IPv6 in an SP network.

6RD

6RD, described in RFC 5969, is an interesting refinement of the 6to4 approach. It shares the same basic encapsulation protocol, and the same address structure of embedding of the IPv4 tunnel endpoint into the IPv6 address. However it has removed the concept of third party relays and the use of the common 2002::/16 IPv6 prefix, and instead uses the provider's IPv6 prefix. The effect of these changes is to limit the scope of the tunnelling mechanism to that of tunnelling across network infrastructure of a single provider, and the intended function is to tunnel from the Consumer Premises Equipment (CPE) to IPv6 Border Relays operated by the customer's ISP.



If 6to4 is not recommended for use because of high failure rates of connections and sub-optimal performance, then why would 6RD be any better?

The most compelling reason to believe that 6RD will perform more reliably than 6to4 is that 6RD removes the wildcard third party relay element from the picture. For outbound traffic the CPE provides the tunnel encapsulation, which is, hopefully, under the ISP's operational control. The IPv6-in-IPv4 tunnel is directed to the ISP's own 6RD Border Relay rather than the 6to4 relay anycast address. As this is also under the ISP's direct operational control, this eliminates the outbound third party relay function. For the reverse path, the use of the provider's own IPv6 prefix in 6RD, instead of the generic 2002::/16 prefix, ensures that the inbound packets are sent via IPv6 directly to the ISP, and the IPv6-in-IPv4 tunnel is again limited to a hop across the ISP's own internal infrastructure.

As long as the ISP effectively manages all CPE devices, and as long as the CPE itself is capable of supporting the configuration of additional functional modules that can deliver unicast IPv6 to the client and 6RD tunnels inward to the ISP, then 6RD is a viable option for the ISP. At the cost of upgrading the CPE set to include 6RD support, and the cost of deployment of 6RD Border Relays that terminate these CPE tunnels, together with IPv6 transit from these Border

Relays, the ISP is in a position to provide dual stack support to its client base from an internal network platform that remains an IPv4 service platform, thereby deferring the process of conversion of its entire network infrastructure base to support IPv6.

For ISPs seeking to defray the internal infrastructure IPv6 conversion costs over a number of years, or for ISPs seeking an incremental path to IPv6 support that allows the existing infrastructure to remain in place for the moment, 6RD can be an interesting and cost effective alternative to a comprehensive dual-stack deployment, as long as the ISP has some mechanism to load the CPE with IPv6 support and 6RD relay functionality.

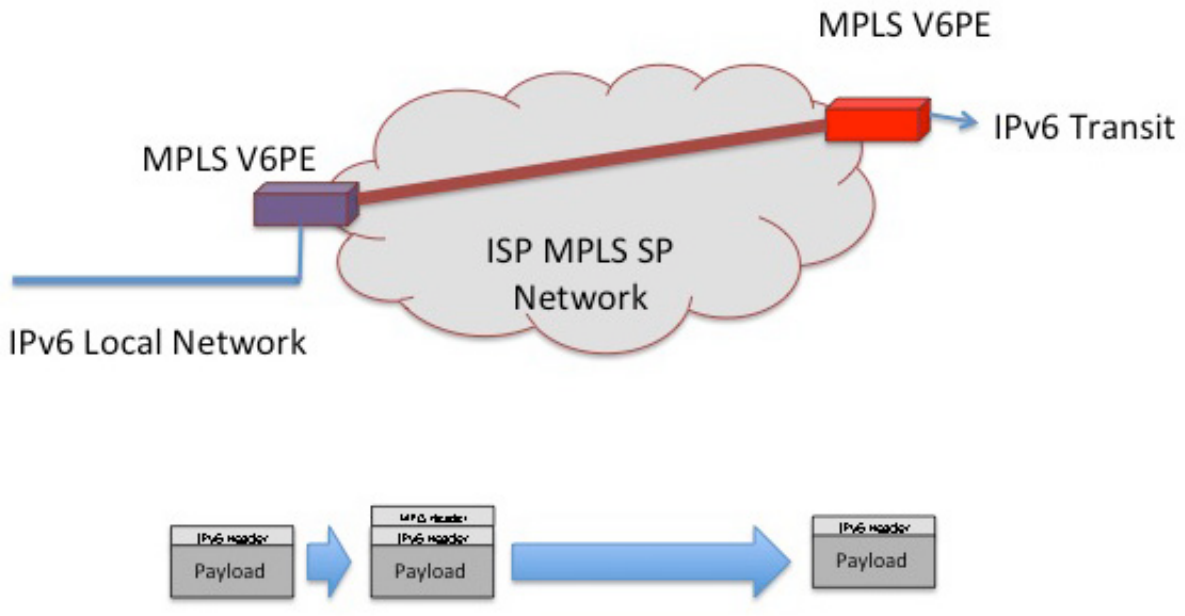
MPLS and 6PE

The 6RD approach has many similarities to MPLS, in that an additional header is added to incoming packets at the network's boundary, and the encapsulation effectively directs the packet to the appropriate network egress point (as identified by ingress), where the encapsulation is stripped and the original packet is passed out.

Rather than using an IPv4 header to direct a packet from ingress to egress, if the network is already using MPLS, why not simply support IPv6 on an existing MPLS network as a PE-to-PE MPLS path set and bypass the IPv4 step?

Why not indeed, and RFC 4659 describes how this can be achieved.

If you are running an MPLS network, then the role of the interior routing protocol and label distribution function is to maintain viable paths between all network ingress and egress points. The protocol-specific function in such networks is not the interior network topology management function, but the maintenance of the mapping of egress to protocol-specific destination addresses.



As with 6RD, if the local issue is some form of prohibitive barrier to the immediate deployment of IPv6 in a dual stack configuration across the network infrastructure, then this approach allows an IPv4 MPLS network to set up paths across the network's IPv4 MPLS infrastructure from PE to PE. These paths may be used to tunnel IPv6 packets across the network, by associating the IPv6 destination address of the incoming packet with the IPv4 address of the egress router, using the iBGP Next Hop address, for example.

The incremental change to support IPv6 are constrained to adding IPv6 to the SP's iBGP routing infrastructure, and to the PEs in the MPLS network, while all other parts of the SP's service platform can continue to operate as an MPLS IPv4 network for the time being.

IPv4 Address Compression

It's not just the challenge of sliding in a new protocol onto the existing IPv4 network infrastructure that is confronting ISPs. The entire reason for this activity is the prospect of exhaustion of supply of IPv4 addresses. When this prospect was first aired, back in 1990, it was assumed that the Internet would supported by industry players that acted rationally in terms of common interests. One of the more critical assumptions made in the development of transitional tools was that transition would be an activity that would be undertaken well in advance of IPv4 address exhaustion. Competitive interest would see each actor making the necessary investments in new technologies to mitigate the risks of attempting to operate a network in an environment of acute general scarcity of addresses. As much fun as the debate as to whom should the "last" IPv4 address be given to might be, it was assumed that this was in fact never going to happen. The assumption was that industry actors would anticipate this situation and take the necessary steps to avoid it. The transition to IPv6 would be effectively complete well before the stocks of IPv4 addresses had been exhausted, and IPv4 addresses would be an historical artefact well before we needed to delve gown to the bottom of the IPv4 address barrel to pull out the very last one!

Obviously, this has not happened.

This industry is going to exhaust the available supplies of IPv4 addresses well before the transition to IPv6 is complete, and in come cases well before the transition process has even commenced! This creates an additional challenge for ISPs and the Internet, and raises a further question as well. The challenge is to fold into this dual stack transition the additional factor of having to work with fewer and fewer IPv4 addresses as the transition process continues. This implies that the necessary steps that the ISP has to take include steps that increase the intensity of use of each IPv4 address, and wherever possible substitute private use IPv4 address for public Ipv4 addresses.

The question that this raises is one of guessing for how long this hybrid model of an Internet where a significant proportion of network services and network clients remain entrenched in an IPv4-only world will persist. For as long as such IPv4-only network domains persist, and for as long as these IPv4-only network domains encompass significant service and customer populations, all the other parts of the Internet are forced to maintain residual IPv4 capability and cannot transition their customers and services to an IPv6-only environment. Students of economic game theory may see some rich areas of study in this developing situation.

More practically, for an ISP the question becomes one of attempting to understand how long this hybrid period of attempting to operate a dual stack network with continuing post-exhaustion demand for further IPv4 addresses will last. Will an after-market for the redistribution of addresses emerge? How will the increasing scarcity pressure impact on pricing in such a market? How long will demand persist for IPv4 addresses in the face of escalating price? Will the industry turn to IPv6 in a rapid surge in response to cost escalation for additional IPv4 addresses, or will a dual stack transition lumber on for many years? In such a large, diverse, heterogeneous environment of today's Internet the one constant factor there is that the immediate future of the Internet is clouded with extremely high levels of uncertainty.

The cumulative effect of the individual decisions taken service providers, enterprises, carriers, vendors, policy makers and consumers has created a somewhat chaotic environment that adds a significant level of uncertainty and associated investment risk into the current planning process for ISPs.

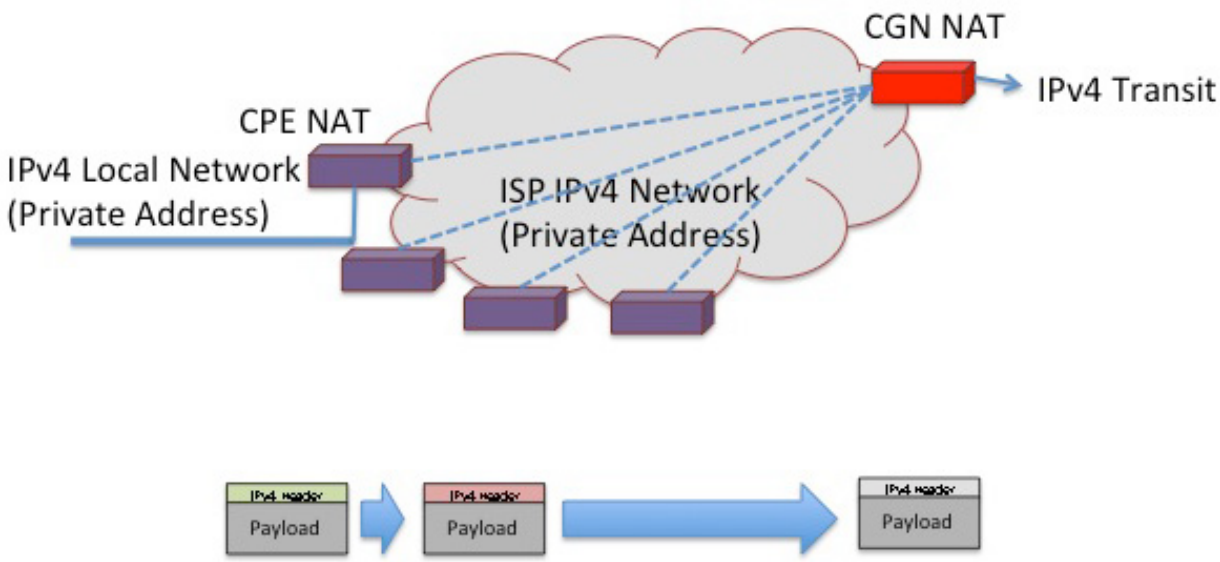
Carrier Grade NATs

I have often heard it said that address scarcity in IPv4 is nothing new, and it first occurred at the point in time when the first Network Address Translator that supported port mapping was deployed. At this point the concept of address sharing was introduced to the Internet, and, from the perspective of the NAT industry, we haven't looked back since!

In today's world NATs are extremely commonplace. Most clients are provisioned with a single address from their ISP, which they then share across their local network using a NAT. Whether its well advised or not NATs typically form part of a client's network security framework, and often are an integral part of a customer's multi-homing configuration if they use multiple providers.

But in this model of NATs as the CPE the ISP uses one IPv4 address for each client. If the ISP wants to achieve greater levels of address compression then its necessary to share a single IPv4 address across multiple customers.

The most direct way to achieve this is for the ISP to operate their own NAT, variously termed a "Carrier Grade NAT (CGN)" or a "Large Scale NAT (LSN)" or "NAT444". This is the simplest of approaches, and, in essence, is a case of "more of the same".



The CGN NAT allows a single public address to be shared across multiple clients, who, in turn, further share this address across the end systems in their local network.

From behind the CPE in the client edge network 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 NAT "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 the CGN 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 associated 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 as moving traffic from one NAT-connected external service to another will cause all the current sessions to be dropped. 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 a CGN is deployed, the same behaviour will consume port binding space on the CGN and, potentially, can starve the CGN of external address port 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 NATed 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.

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?

The Address plus Port Approach

One NAT in the path is certainly worse than none from the perspective of application agility and functionality. And two NATs does not make it any better! Inevitably, that second NAT adds the some additional levels of complexity and fragility into the picture.

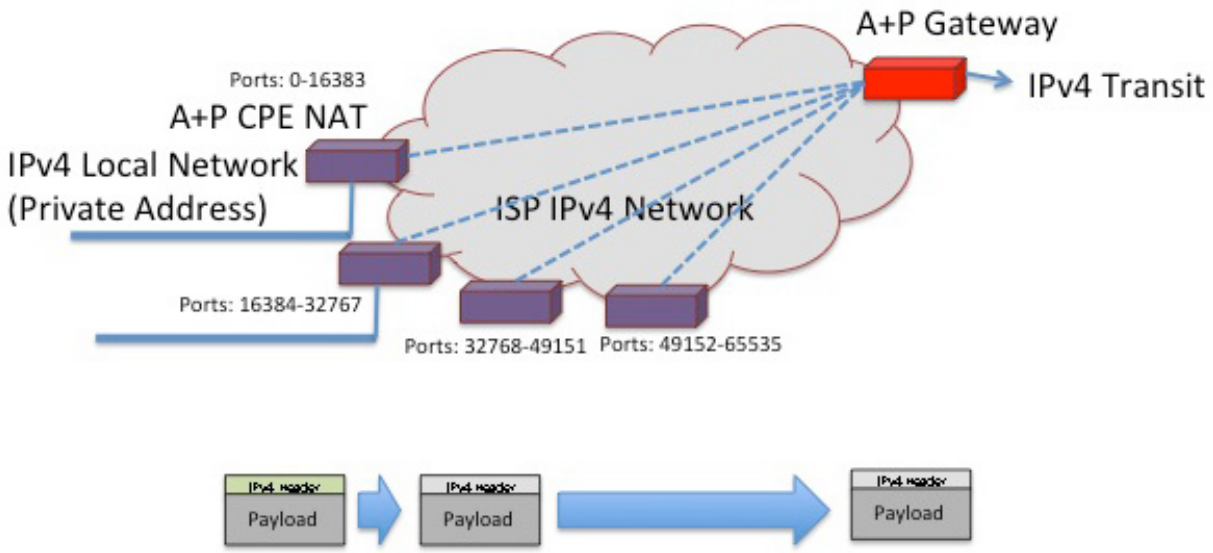
The question is, can these two NATs be collapsed back into a single NAT, yet still allow sharing of public IPv4 addresses across multiple end clients? CPE NATs currently map connections into the 16 bit port field of the single external address. If the CPE NAT could be coerced into performing this mapping into, say, 15 bits of the port field, then the external address could be shared between two edge CPEs, with the leading bit of the port field denoting which CPE. Obviously, moving the bit marker further 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 range of external port values it can use in its NAT bindings to those that lie within the provided port range.

Other CPE devices are concurrently using the same external IP address, but 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. The CPE is then constrained such that as it maps private addresses and TCP/UDP port values to the external address and port values, the mapped port value must fall within the configured range.

The handling 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, as there are now multiple CPE egress points that share the same IP address. 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 A+P gateway. This A+P gateway 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 that are similarly tunnelled from the CPE to the A+P gateway). Incoming packets are encapsulated in IPv6 using the IPv6 destination address that it has learned previously. In this manner the NAT function is performed just once, at the edge, much as it is today, and the interior device is a more conventional form of tunnel server.



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 and egress functions, and act as an IPv6 provisioned endpoint for the SP network. This is perhaps an unrealistic set of constraints for may SP networks. 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 A+P functionality.

This approach has some positive aspects. Pushing the NAT function back to the network's 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

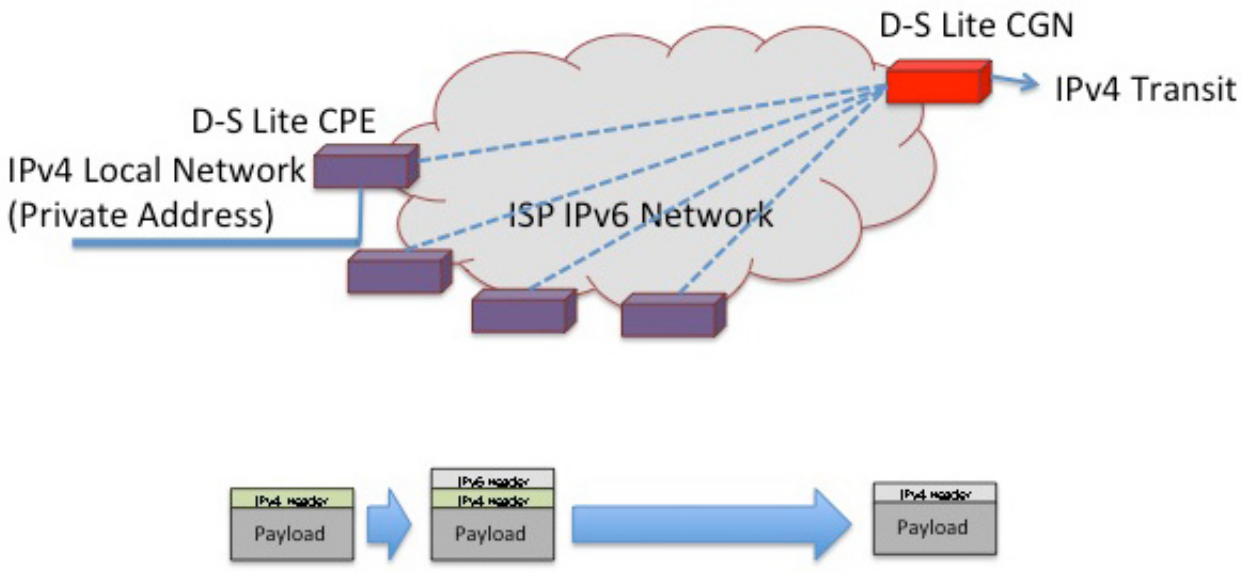
means of address sharing, given that once a CPE has been assigned a port range, those port addresses are unusable by any other CPE. The prudent SP would assign to each CPE a port address pool equal to some estimate of peak demand, so that, for example, each CPE would be assigned some 1,024 ports, allowing a single external IP address to be shared across only some 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 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.

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.

Dual-Stack Lite

The concept behind the Dual-Stack Lite 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.

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 interior 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 D-S Lite NAT fails in any way then the entire customer base is disrupted. It seems somewhat counter-intuitive to create a resilient end-to-end network with stateless switching environments and then place a critical stateful unit right in the middle!

Protocol Translation

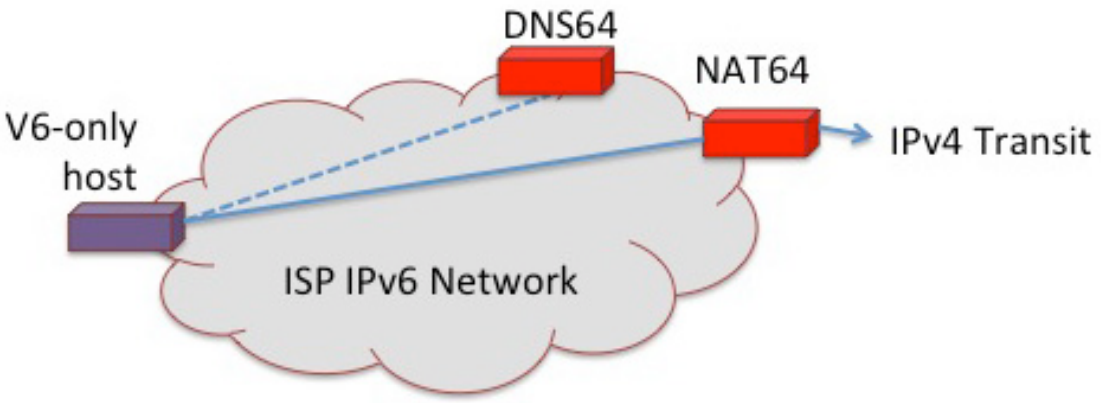
So far we've looked at two general forms of approach to hybrid networks that are intended to support both IPv6 transition and greater levels of address utilization in IPv4, namely address mapping and tunnelling. A third approach lies in the area of protocol translation.

RFC2765 contains the details of a relatively simple protocol translation mechanism. The approach relies on the basic observation that IPv6 did not make any radical changes to the basic IP architecture of Ipv4, and that it was therefore possible to define a stateless mapping algorithm that could translate between certain IPv4 and IPv6 packets. Of course the one major problem here is that there are far far more addresses in IPv6 than IPv4, so the approach used was to map IPv4 addresses into the trailing 32 bits of the IPv6 address prefix ::FFFF:0:0/96. The approach assumed that to the IPv6-only end host the entire Ipv4 network was visible in this mapped IPv6 prefix, and that when the Ipv6-only end host wished to communicate with a remote host who was addresses using this IPv4-mapped prefix it would use a source address also drawn from the same IPv4-mapped prefix. In other words it assumed that all IPv6-only hosts were also assigned a unique IPv4 address.

The NAT-Protocol Translation (NAT-PT) approach attempted to relax this constraint, allowing IPv6-only hosts to use a dynamic mapping to a public IPv4 address through the NAT-PT function, in the same way as NATs work in an all-IPv4 domain. The proposed approach assumed that the local host was located behind a modified DNS environment where the IPv4 A record of an IPv4-only remote service is translated by the DNS gateway into a local IPv6 address where the initial 96 bits of the IPv6 address identify the internal address of the NAT-PT gateway and the trailing 32 bits are the IPv4 address of the remote service. When the local host then uses this address as an IPv6 destination address, the packet is directed by the local routing environment to the NAT-PT device. This device can construct an "equivalent" IPv4 packet by using the local IPv4 address as the source address, the last 32 bits of the IPv6 address as the destination address, and bind the IPv6 source port to a free local port value. These set of transforms can be locally stored as an active NAT binding. Return IPv4 packets

can be mapped back into their "equivalent" IPv6 form by using the values in the binding to perform a reverse set of transforms on the IP address and port fields of the packet.

This approach was published as RFC2766 in February 2000. Some 7 years later in July 2007, the IETF published RFC4966, deprecating NAT-PT to "historic", with an associated laundry list of applications which would not operate correctly through such a device. This negative judgement of NAT-PT seems rather curious to me, given that conventional CPE NATs in IPv4 appear to share most, if not all, of the same shortfalls that are listed in RFC4966. Given the extensive set of compromises that are required in the environment that is partially crippled by IPv4 address exhaustion, it seems rather contradictory to insist upon extremely high levels of functionality and robustness from these hybrid translation approaches.



Not unsurprisingly, NAT-PT is undergoing a revival, this time under the name "NAT64." Not much has changed from the basic approach outlined in NAT-PT. The IPv6-only client performs a DNS lookup through a modified DNS server that is configured with DNS64. IN the case that the queried name only contains an IPv4 address, the DNS64 server synthesises an IPv6 response by merging the prefix address of the NAT64 gateway with the IPv4 address. When the client uses this address, the IPv6 packet is directed to the NAT64 gateway, and the same transform as described above for NAT-PT takes place.

This setup is similar to the CGN model, in so far as the service provider operates a common NAT that shares an IPv4 address pool across a set of end clients.

Conclusions

There really is no single clear path forward from this point. Different ISPs will see some advantages in pursuing different approaches to this dual problem of introducing IPv6 into their service portfolio and at the same time introducing additional measures that allow more efficient use of IPv4 addresses.

However, one common theme is becoming clear. So far ISPs have been able to 'externalise' many of these issues by pushing much of the complexity and fragility of NATs out to the customer and loading up the CPE with this functionality. This approach of externalising much of the complexity of address compression in NATs over to the customer's network cannot be

sustained with the IPv6 transition, and no matter which approach is used, whether it's a CGN, NAT64, Dual-Stack Lite, 6RD or MPLS with 6PE, the ISP now has to actively participate in the delivery of IPv6 and in increasing the efficiency of use of IPv4.

So for the ISP its time to start making some technical choices as to how to address the combination of these two rather unique challenges of transition and exhaustion.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. 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.

www.potaroo.net