The ISP Column
A monthly column on things Internet

September 2008

Geoff Huston

IPv6 Transition at IETF72

The IETF's developmental work on IPv6 has included the study of the particular issues associated with transition to IPv6 from the outset. The first initial effort in the transition space, at IETF29 in March 1994, was termed "TACIT", an acronym of Transition and Coexistence including Testing. While this was admittedly a forced acronym, it was illustrative of the IETF's desire to include consideration of transition issues as part of the design of IPv6 itself. The underlying consideration here is a study of how a diverse collection of applications, hosts and network elements that collectively make up the Internet and the related collection of enterprise networks can be upgraded, selectively augmented or replaced in order to support IPv6 and, ultimately, to deprecate all further use of IPv4 while at the same time preserving all the essential any-to-any end-to-end property of IP through the transition. From the TACIT BOF sessions the baton was then passed to the NGTRANS Working Group in July 1995 at IETF33. This working group was active until mid-2002 with IETF 55, when the baton was again passed over, this time to the V6OPS Working Group which met first in early 2003 at IETF56. This study has now broadened in scope and today a number of IETF working groups are examining aspects of transition to IPv6 including the SOFTWIRE, BEHAVE and INTAREA Working Groups, in addition to V6OPS.

Given that this study now encompasses a period of 14 years, what exactly are the issues with transition to IPv6, and why is it taking such a long time?

A Classical Transition

Perhaps the first observation to make is that IP and an end-to-end protocol, as distinct from a hop-by-hop protocol. The approaches used in the transition in BGP from 16 bit AS numbers to 32 bit AS numbers used a combination of translation and tunnelling that effectively allowed a BGP speaker that was configured to use the longer AS numbers to be "backward compatible" with the existing installed base of BGP that uses the earlier 16 bit AS number format. However this backward compatible translation technique relies on the properties of hop-by-hop interpretation of tokens, where AS number values are interpreted in a strictly local context. This does not apply to IP where a packet's destination IP address needs to have meaningful context at all points in the network, and there is no a priori constraint on the interpretation of an IP address. All IP addresses need to be meaningfully interpreted in all parts of the network al all times. This implies that the use of translation and substitution has limited applicability in the context of IP addresses.

The original approach to IPv6 deployment was a "classical" view of transition. Because IPv6 is not a backwards compatible augmentation of IPv4 it is not possible to deploy new hosts and network infrastructure with support for only IPv6 and have these networks, devices and applications exchange IP packets with their IPv4 counterparts. An application that is equipped with IPv6 requires its host to have IPv6 support in its protocol stack, and for the host to be able to communicate it requires the network to have IPv6 support. And if an application wishes

to communicate with another application all the networks on the path between the two hosts also must be configured to support the transmission of IPv6 packets. In other words a "complete" deployment of IPv6 requires all applications, hosts, and network infrastructure and middleware to be aware of IPv6 and explicitly configured to handle IPv6 packets. In this "classical" form of transition the major constraint is to avoid any flag day, or any form of synchronized or orchestrated common activity across the entire network. Individual elements of the network should be able to undertake their part of the transition without requiring any action to be performed on any other element. The transition should be a piecemeal activity. This "classical" approach, in general terms, assumes that each application, host device and network element is able to make an independent decision as to when to enable support for IPv6. In order to preserve connectivity of the network as a whole as and when each element added support for IPv6 it would not "cut over" and remove all IPv4 support, but, instead, it would support the operation of both IPv4 and IPv6 for a period. This was termed the "Dual Stack" transition approach. This mode of progressive shift of the elements of the Internet to a Dual Stack operation would continue for as long as there were essential components of the overall environment, from applications to internet infrastructure that support only IPv4. Only when the entire connectivity domain was supporting comprehensive Dual Stack operation would it be possible to deprecate IPv4 from the network and remove all support for this protocol.

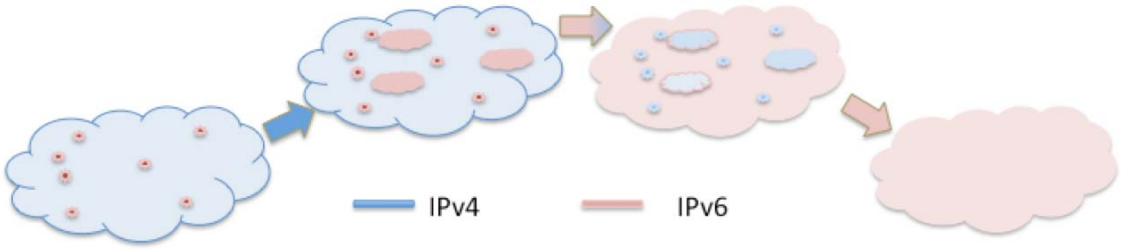

Figure 1 - The Progressive Stages of IPv6 Transition

The issue with this approach to IPv6 transition is that it relied on a strong mix of altruism, common purpose, shared motivation and a high level of technical capability from everyone, from suppliers and vendors through to network operators and even end users. For the early adopters of IPv6, whether it was application designers, suppliers of host operating systems or routers, or network operators and system administrators, the investment in Dual Stack capability in their area of responsibility would only generate the greatest extent of resultant benefit when the transitional Dual Stack phase was complete. In other words, there was no immediate reward for those early adopters of IPv6, and the late adopter did not experience any detrimental side effects, as the full benefits of an outcome of IPv6 adoption would only be realized once the entire environment adopted IPv6 in a Dual Stack configuration with IPv4, at which point IPv4 could be deprecated from the operational network.

This approach assumes that all parties are equally motivated to undertake this transition, and for each party to do so as quickly as possible. It also assumes that all applications, all connected devices, and all components of the network's infrastructure are capable of being configured to operate in Dual Stack mode. Perhaps these assumptions may have been feasible in practical terms if IPv6 had been in a position to offer very significant cost, performance or functionality improvements over IPv4. In such a case the superior characteristics of the new technology would've propelled the transition process in any case. However, any such major relative improvement in performance, cost and utility is not the case when comparing IPv6 to IPv4, as IPv6 represents only a marginal change in the underlying packet design. Following a further decade of incremental refinement in both IPv4 and IPv6 we have the current situation that, apart from the larger address fields in the packet header, there is no significant relative change in IPv6 from a performance or benefit perspective. In addition, the Internet itself is now so much larger and so much more diverse that commonality of purpose is difficult to

sustain. These days altruism often takes a back seat to business interests, as the Internet now operates as a collection of quite conventional business enterprises. Indeed since the bursting of the Internet bubble at the start of this decade, this sector of business is relatively conservative as well, and a far greater emphasis is placed on securing immediate returns on invested capital over and above undertaking longer term investments with less certain outcomes. This implies that any such commonality of purpose and a vision of a longer term outcome is extremely challenging to sustain in the face of shorter term considerations.

The combination of these factors create a situation that has been incapable of sustaining the operation of this "classical" transition process. So the IETF was motivated to look at transition in slightly different terms, to see if this approach could be refined to offer some more immediate benefits to early adopters and not to stall the entire process awaiting the completion of the late adopters of Dual Stack.

Transition with Incremental Outcomes - Tunnelling

The initial refinement to this original transition model, explored in the NGTRANS Working Group, was intended to allow various IPv6-only and Dual Stack applications to support IPv6 from the outset, so that any benefits related to IPv6 could be realized immediately, and not be forced to await the actions of the slowest adopters to also make their moves. The motivation related to the restoration of simple APIs for applications, the restoration of coherent end-to-end packet delivery in an IPv6 network, and the benefits that this clear and simple application architecture offers to applications that operate in an "over-the-top" mode. Such an end-to-end packet transport environment offers strong end-to-end channel security, as well as restoration of the uniform binding of IP address to end-point identity in the IP architecture.

The objective of attempting to operate in and end-to-end IPv6-only mode over a largely IPv4 substrate network lead to the development of a number of approaches to IPv6 transition that relied on tunnelling techniques, where IPv6 packets are encapsulated in an IPv4 packet wrapper, allowing these IPv6 "islands" to treat the IPv4 network as a form of transmission media, or a non-broadcast multicast network. This has lead the development of the general technique of carrying IPv6 packets in IPv4 by treating IPv6 as an IPv4 protocol, namely protocol 41.

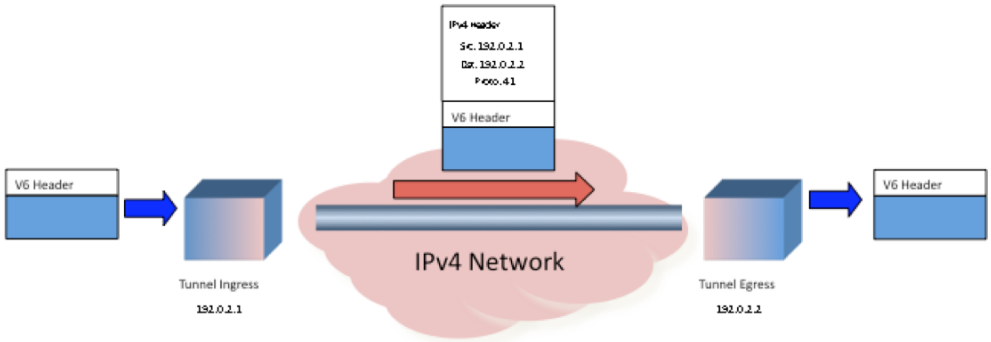

Figure 2 - IPv6 in IPv4 tunnelling

The general characterisation of this approach to this form of Dual Stack transition was to allow the initial "islands" of IPv6 adoption to connect to each other via these tunnels, essentially creating an IPv6 connected network from the outset. As more of the infrastructure adopted the same form of Dual Stack support these "islands" would start to directly interconnect, making the "islands" larger and the tunnelled "gaps" shorter. As these gaps shrink to the point of general Dual Stack support it may be an option to then tunnel the remaining IPv4 traffic over IPv6, but perhaps that's getting well ahead of ourselves right now.

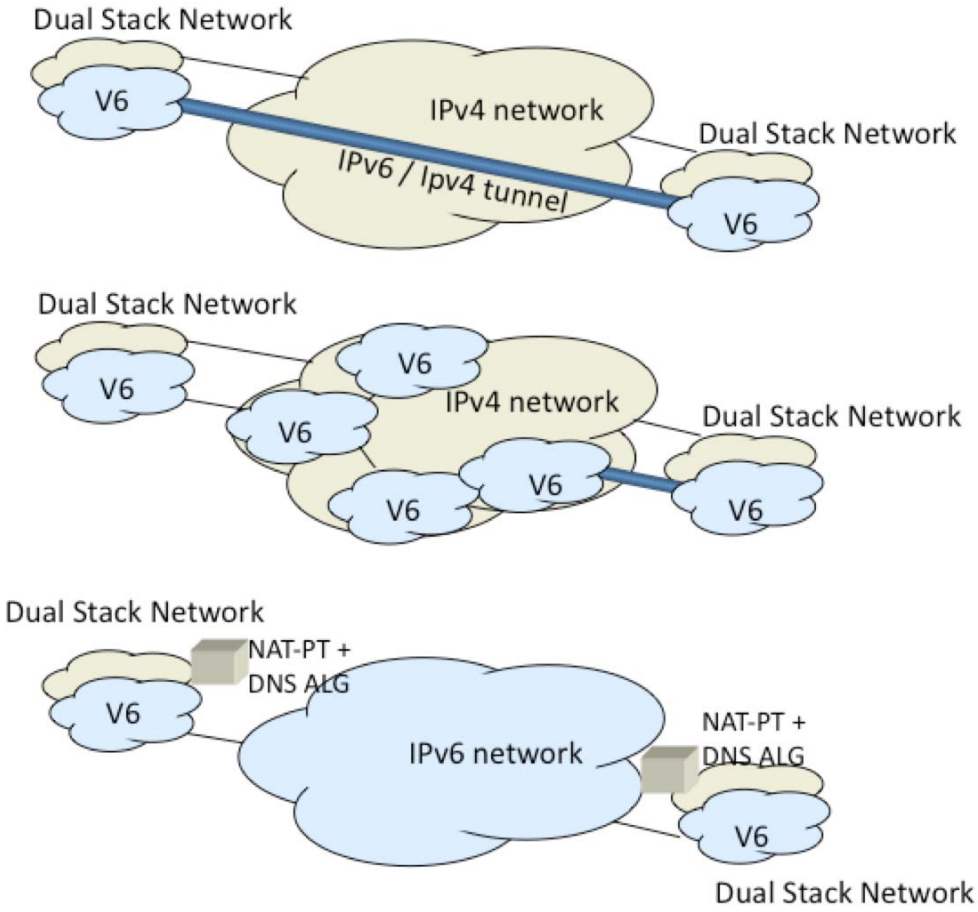

Figure 3 - Transition using Tunnels

While the motive and logic for the use tunnels in this transition scenario is certainly sound, the overhead here is that tunnels normally require explicit configuration of both "ends" of the tunnel, and any form of tunnel topology that attempts a fully meshed interconnection of these IPv6 "islands" runs into a N-squared scaling problem in tunnel configuration almost immediately.

This, in turn, has lead to exploration of approaches that supported the concept of fully meshed tunnels, but with an extremely simple single end configuration. This is achieved by associating an IPv4 tunnel endpoint in an endpoint IPv6 address. When such a packet is passed to a tunnel ingress, the IPv4 tunnel egress address is defined by the original IPv6 destination address, so that the tunnel does not have to be explicitly configured at both ends. One of these is the 6to4 technique, which generates an IPv6 48 bit prefix by prepending 2002::/16 to the front of the 32 bit IPv4 address. This allows a Dual Stack gateway to double as a IPv6 tunnel egress, serving a local network of IPv6 hosts with tunnel services. Each 6to4 gateway, or 6to4 individual host needs only to configure its "end" of the tunnel. All IPv6 packets between 6to4 sites are passed directly from 6to4 gateway to gateway. To complete the picture each local 6to4 network needs to provide 6to4 gateway service for IPv6 packets from non-6to4 IPv6 networks.

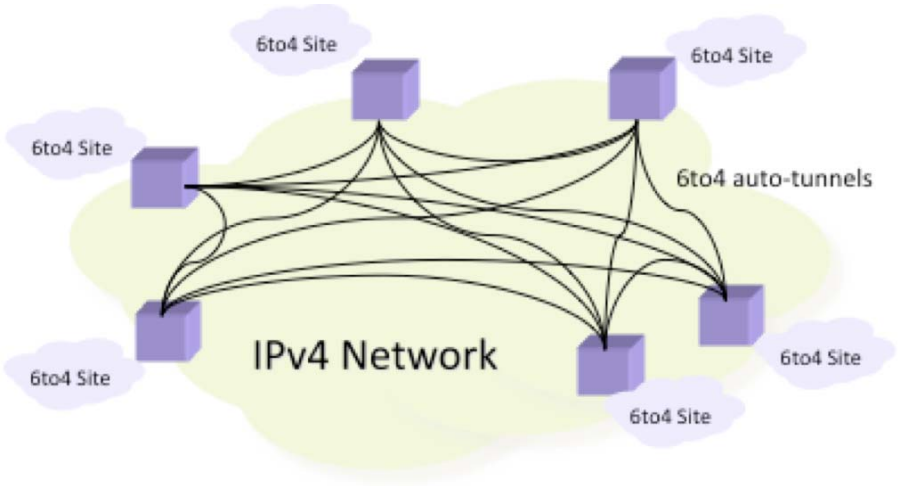

Figure 4 - 6to4 Tunnelling

A related form of embedding IPv4 in IPv6 addresses to aid in auto-tunnelling is ISATAP, the Intra-site Automatic Addressing Protocol, which embeds the IPv4 address in the interface identifier field of the IPv6 address to support a local scope automated IPv6 over IPv4 tunnelling approach. These approaches can be combined, so that an enterprise can construct an IPv6 network with a single infrastructure gateway which creates the prefix and tunnels over the wide area network using 6to4, while tunneling over the local area network using ISATAP.

The shortcoming of the 6to4 approach is that is assumes a general availability and use of public IPv4 addresses. A single host behind a NAT gateway cannot use this approach given that the implicit IPv4 tunnel endpoint is drawn from a private address pool and is therefore not visible outside the IPv4 private address scope. It also requires firewalls to be aware of protocol 41 and apply the IPv6 filter rules to the inner IPv6 packet.

The Teredo approach addresses both these concerns by using explicit support for NAT traversal, and embedding the IPv6 packet inside an IPv4 UDP transport session rather than as an IP transport. Teredo uses a relatively conventional approach to NAT traversal, using a simplified version of the STUN active probing approach to determine the type of NAT, and uses concepts of "clients", "servers" and "relays". A Teredo client is a Dual Stack host that is located in the IPv4 world, possibly behind a NAT. A Teredo server is an address and reachability broker that is located in the public IPv4 Internet, and a Teredo relay is a Teredo tunnel endpoint that connects Teredo clients to the IPv6 network.

The tunnelling protocol used by Teredo is not the simple IPv6-in-IPv4 protocol 41 used by 6to4. IPv4 NATs are sensitive to the transport protocol and generally pass only TCP and UDP transport protocols. In Teredo's case the tunnelling is UDP, so all IPv6 Teredo packets are composed of an IPv4 packet header, a UDP transport header, followed by the IPv6 packet as the tunnel payload. Teredo represent a different set of design trade-offs as compared to 6to4. In its desire to be useful in an environment that includes NATs in the IPv4 path Teredo is a per-host connectivity approach, as compared to 6to4's approach which can support both individual hosts and end sites within the same technology. Also, Teredo is now a host-centric multi-party rendezvous application, and Teredo clients require the existence of Dual Stack Teredo servers and relays that exist in both the public IPv4 and IPv6 networks. From Teredo's host-centric perspective it could be said that Teredo is more of a connectivity tool than a service solution.

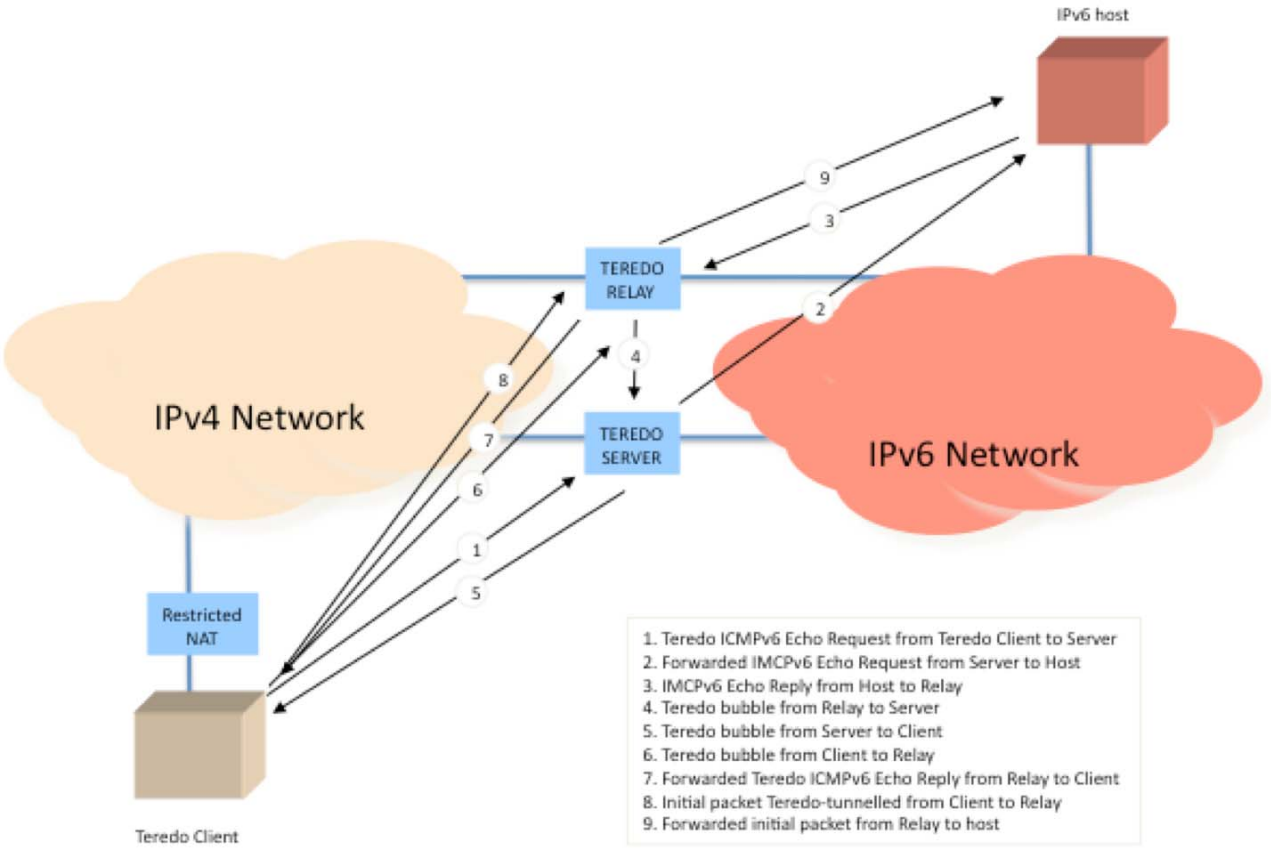

Figure 5 - An Example of a Teredo Rendezvous

The common feature of all of these transition approaches is the use of tunnels. Tunnels are extremely convenient in terms of their ability to interconnect diverse islands of IPv6 without requiring any change to the intervening Ipv4 infrastructure. However, tunnels are not without their attendant problems. Tunnels can be fragile, unstable and challenging to diagnose. The issue of ICMP treatment within tunnels is a good example, where a return ICMP error notice is sent not to the original source host, as intended, but to the tunnel ingress point which is the source address of the outer tunnel packet. The inner payload, which contains the initial fragment of the original packet also includes the tunnel header. The critical point here is the interplay between end-to-end signalling and maximal Message Transmission Unit (MTU) discovery. Where there is a tunnel MTU mismatch coupled with an ICMP handling problem, the situation often manifests itself as a TCP "hang", where the initial SYN handshake succeeds, but the first large data packet is never transmitted. A typical Dual Stack implementation will lock into IPv6 or IPv4 at the point of completion of the initial TCP handshake completion, and the data payload problem then causes the user's application to hang while the name to protocol family association is now locked into the user's cache, so that resetting the connection and forcing the application to use IPv4 rather than IPv6 is invariably beyond the user's direct control.

So is it possible to avoid tunnels and still achieve incremental outcomes for early adopters of Dual Stack? Behind all the transition scenarios so far lies the assumption the IPv4 and IPv6 support distinct 'universes' of connectivity. However, both protocols present much the same set of functions to the upper level transport protocols, and the header fields of the protocol are similar. Just had bad is this backwards incompatibility of IPv6 with respect to IPv4? Is it completely impossible for an IPv4-only host to initiate, maintain and close a "conversation" with a IPv6-only host and vice-versa? If one allowed various forms of intermediaries, including protocol-translating NATs and various permutations of DNS servers, is this still impossible? Probably not impossible, but it would go well beyond the conventional mode of packet protocol header substitution, and would call upon protocol header translation, cross protocol NAT bindings, DNS manipulation and various forms of application level gateways.

An approach to this form of translation was described in RFC2766, “Network Address Translation - Protocol Translation (NAT-PT)”. The approach creates a number of security vulnerabilities and appears to operate with a high level of assumption about application behaviours, making its operation extremely fragile. The NAT-PT approach subsequently deprecated in a more recent RFC, RFC 4966, which consigned NAT-PT from Proposed Standard to Historic status, with the comment that: “Accordingly, we recommend that: the IETF no longer suggest its usage as a general IPv4-IPv6 transition mechanism in the Internet, and RFC 2766 is moved to Historic status to limit the possibility of it being deployed inappropriately.”

IPv4 Exhaustion and IPv6 Transition

The one common assumption in all these transition scenarios is that this Dual Stack transition will take place across the period when there was still sufficient IPv4 addresses to address the entire Internet across the entire transition phase., and that the event that IPv6 was primarily intended to avert, namely the exhaustion of the supply IPv4 addresses from the unallocated pool, would not occur during the transition process.

It is generally anticipated that this transition will take up to a further decade to complete from the current time, while depletion of the unallocated IPv4 address pool may occur within the next two to three years. While the overall transition tool box always assumed a wide array of deployment approaches, this forecast shortage of IPv4 will shift the scaling trade-offs for transition approaches in ways that will be more complex and expensive to operate than the simpler dual-stack approach would have been. On the other hand, this is a forced scenario as there is no opportunity to go back in time to try this transition again under different circumstances.

Whatever scenario of IPv6 transition we contemplate, it now has to be one that will take into account the forthcoming acute shortage of public IPv4 addresses, which implies an environment that is heavily reliant on various forms of NATs and possibly some further extensions to NAT behaviours and NAT deployment models, including the possibility of augmenting the "NAT at the edge" deployment model with various forms of "NAT in the middle," as the industry contemplates the potential of so-called "Carrier Grade NATs" and related approaches.

The challenge as we undertake these new technical approaches will be to not lose sight of the fact that short-term cost pressures need to be balanced against the collective long term desirable outcome of an achievable exit strategy from the ever more complex environment of keeping IPv4 operating.

IETF 72 Activity

In IETF 72 the issues that we have in confronting with this combination of Dual Stack transition to IPv6 and IPv4 address depletion were discussed in a number of working groups, as well as the Technical Plenary session. What follows here is a very brief summary of the relevant activity in each of these working groups. While these brief summaries provide a general overview of current activities, the brevity of the description here can get in the way of precision, and the reader is referred to the proceedings of the IETF72 meeting, and of course the associated internet drafts for a more complete description of these technical contributions (http://www.ietf.org).

At the Technical Plenary the IETF was shown some of the underlying metrics of address allocation and the current predictions of depletion of the unallocated IPv4 address pool in 2011. The prospect of broadening the domain of NAT deployment from the edges of the network to parts of the interior boundaries using so-called "carrier Grade NATs" was also foreshadowed at this session. A report of the experience gathered at Google pointed to a pragmatic approach to Dual Stack deployment that advocated undertaking IPv6 support designed to the same

production quality standard as IPv4. It was reported that Google was not in a position to Dual Stack their major service point at present, given that IPv6 today still represents lower reliability and higher latency for some users as compared to IPv4 connectivity to the same service point. A presentation from Apple pointed to consumer products that already make use of IPv6 Link-Local Addressing. This presentation also looked at a dataflow model of connection establishment in a Dual Stack environment, where both IPv4 and IPv6 connections are initiated in parallel, and the first path to successfully complete the DNS and initial packet exchange to complete the connection is the protocol that is associated with the application's original connection request.

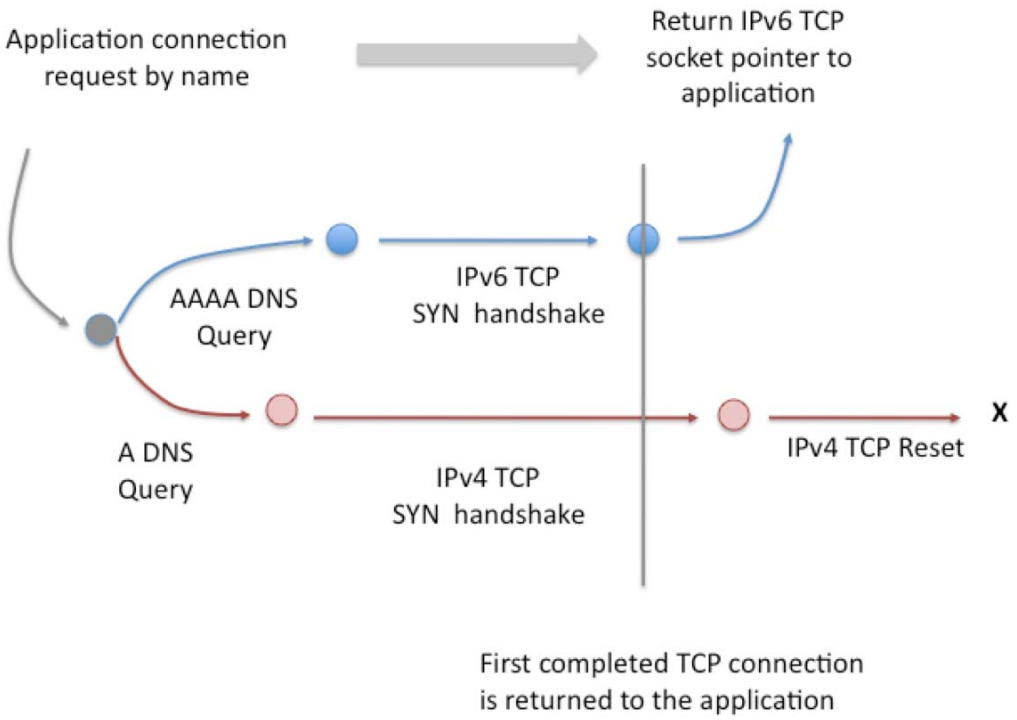

Figure 6 - A Dataflow model of Protocol Selection [Adapted from: Stuart Cheshire, Apple, IETF 72 Plenary]

The V6OPS Working Group is looking at some of the basic operational tools to support transition, and, in particular, a re-examination of the requirements for V4-V6 translation mechanisms to see if there are viable approaches to provide the original NAT-PT function that might address some of the shortcomings in the original specification. The basic problem being addressed by this effort can be envisaged in a scenario where there are no more IPv4 addresses and a network domain is deployed using only IPv6, and this domain wants to be able to communicate with a domain which is still operating in IPv4 only and has not deployed IPv6. At IETF 72 the Working Group reviewed a set of goals to see if there could be a viable set of requirements that could be refined from such a set. While this approach of first defining a set of requirements and then working on potential solutions is a conventional mode of operation for the IETF, the note relating to market timing, where deployment of a solution is anticipated to be needed by 2010 is a very sobering call to focus the effort here. A related effort concerns the evaluation of modified NAT behaviour where the conventional binding space of a vector of "inner" and "outer" addresses and ports and an associated protocol is replaced by an outer side address and port and an inner side tunnel identifier and an address and port that refers to the NAT device at the other end of the tunnel. The essential concept here is that the NAT function is then a distributed function across a common "outer" facing edge device and the set of inner NATs that are used as CPE device. Other work presented at IETF 72 includes a review of proposed refinement of the Teredo specification that would improve its NAT behaviour discovery function from the simple two mode discovery in the current specification to a mode that discovers up to 8 different NAT types. The motivation here is that

the more Teredo traffic that can be offloaded from the Teredo relay to an optimised peer-to-peer connection the more reliable the Teredo performance. Related work has been re-examining the security issues that are exposed by the use of tunnelling and the potential for disruption and hostile attack on the tunnel.

The BEHAVE working group started out with a charter to provide some standard specifications for the behaviour of IPv4 to IPv4 NAT units, but in recent times this has been expanding to encompass the examination of the role of NATs in various IPv6 transition scenarios., including the examination of NATs that perform protocol translation. The current agenda of contributions to review includes the IVI scheme, a proposal to use bidirectional address mapping between subsets of IPv6 and IPv4 addresses to allow a form of stateless transition where the "binding" of the translation is carried in the address fields of the packet itself. Another approach to NAT-PT is also being studied. In this case the asymmetric nature of conventional 4-to-4 NATs is exploited here and a proposal for a 6-to-4 NAT was made to the working group. In this contribution the communication is initiated by the IPv6 host and the synthesised view of the remote IPv4 world is provided by embedding the IPv4 address in the synthesised IPv6 address. The NAT64 host performs a protocol translation by extracting the IPv4 address out of the IPv6 destination address, and providing one of its own addresses as the source address of the IPv4 packet. A NAT binding state is maintained, indexed by the IPv4 address values. The reverse packet performs a binding lookup, allowing the IPv6 destination address to be substituted, and the source IPv4 address is again wrapped up in the synthesised IPv6 packet. BEHAVE also reviewed a contribution calling for the specification of the so-called "Carrier Grade" NATs, where the NAT translation function is provided at the interior boundary of an ISP network, in conjunction with NATs being performed at the CPE edge.

The SOFTWIRES working group has also been involved in aspect of the IPv6 transition, with the consideration of the Softwares NAT, or SNAT. SNAT combines IPv4 NAT and IPv4-in-IPv6 softwires to carry IPv4 traffic through the ISP network that uses only IPv6 service. In essence this approach creates a "split" NAT where the "inner" NAT is connected to the "outer" NAT via a IPv6 software tunnel. Multiple CPE NATs are multiplexed through a single external NAT, reducing the total number of IPv4 addresses in use by the ISP.

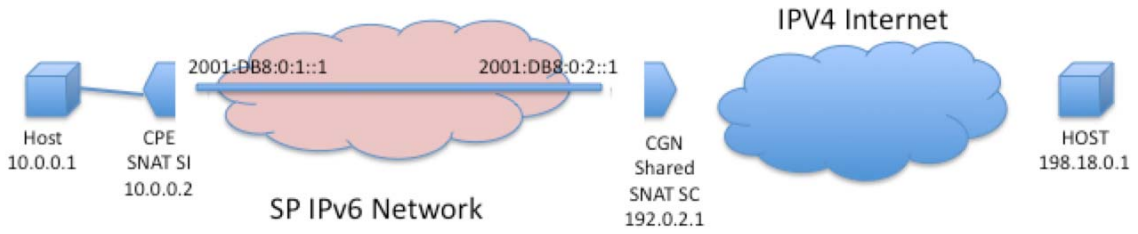

Figure 7 - Softwires SNAT

The INTAREA meeting considered a proposal that calls for standard handling of MTU negotiation, fragmentation and signalling for tunnels. Given that tunnels appear to be a major component of this piecemeal IPv6 transition model the consistent treatment of tunnelled traffic appears to be an emerging near term imperative for the transitioning Internet, and the IPv6 Internet as well. The impending exhaustion of the IPv4 address pool has caused another "critical use" address proposal to emerge. In this case it's a call for a reservation of IPv4 unicast address space to be used within a carrier's infrastructure to bridge the gap between the Carrier Grade NAT at the boundary of the carrier's network and the CPE devices at the boundary to the customer. Given the often protracted debates such calls for reservation of address space often engender, and the relatively short timeframe left for the exhaustion of the remaining pool of IPv4 addresses its not clear if the IETF will be able to reach a clear consensus on this proposal in the remaining time available.

Summary

There is no doubt that the impending exhaustion of the IPv4 unallocated address pool adds some level of urgency as well as an element of complexity to the IPv6 transition agenda, and the work in the IETF will no doubt increase in intensity in the coming meetings. We appear to be now working under quite strict time pressures in developing the standard specification for tools and protocol mechanisms that need to be fielded into production networks within a very compressed timeframe, and having every vendor, every network operator and every operating system suppler devise distinct approaches runs the risk of making the situation more difficult than it would otherwise be.

The challenge for the IETF is to ensure some clarity of focus on the work around transition tools for IPv6 that also assists in increasing the address utilization efficiency of IPv4 addresses, and be mindful of the increasingly strident call of standardization of these hybrid technologies that couple tunnelling, address mapping, NATs and protocol transformation in ways that application designers, operating system vendors and providers and operators of networking infrastructure can use in simple and effective ways.

Disclaimer

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