IPv6 Implementation Lessons

Ian Gray, Savant Ltd
Version 1 - November 2010
Version 2 – January 2015

Preamble

As indicated above I originally wrote this document in 2010. Between then and 2014 not a lot has changed. The observations made in the original document remain generally true. Some lessons have been learned and these have influenced minor parts of the text. What is clear is that Internet providers in the UK generally show very little interest in offering IPv6 particularly to domestic customers (there are around 4 listed on the http://www.thinkbroadband.com/isps.html web site at the time of writing although larger providers of leased line and high capacity circuits may well be excluded from this list). Whether this is due to ignorance or cost is unclear. Further experience in deploying aspects of IPv6 has indicated that even network equipment providers are still developing their offerings since I have uncovered inconsistencies between my understanding and what has actually been delivered.

Introduction

It is almost easier to say what this document is not rather than what it is. Essentially this is about observations, lessons and discussions around the implementation of IPv6 on a small to intermediate network.

It is not a manual or definitive work on how to manage the deployment of IPv6 and it is certainly not a raison d'être for why you should be planning such an implementation. It has become clear as I rolled out IPv6 infrastructure that for all the "IPv6 Ready" claims attached to devices and software there is a great deal of marketing hype and box ticking going on. So instead of being behind the wave of deployment where all the usual questions have an answer somewhere on the Internet I find that I am more leading edge than I had anticipated. Which is strange considering how much of a sales opportunity you would have thought a new protocol would have been attractive to the, not generally reticent, computer and network industry.

So who am I or more particularly the company I work for. Savant, the company, is a 40 person software company specialising in healthcare solutions mostly in relation to blood transfusion and haematology. We rely on good communications between our systems and require reliable connection to the Internet where a great number of our support questions are answered. We are not, therefore, a large company and have some flexibility in developing and trying new technologies that may be missing from larger organisations.

This then is the background to the deployment. The following is a ramble through the actual mechanisms and blind alleys that were encountered along the way.

The Easy Wins

There are none. The only thing you are buying by implementing IPv6 is the future proofing of your network and the knowledge that when the IPv4 address space runs out you won't be the headless chicken that suddenly has to design and build a new infrastructure in a short period of time. It is my guess, and it is only a guess, that it is those organisations that trade with the far east, where IPv6 is more pressing and more advanced, that will see the need to change first. Remember that 60% of the IPv4 space is assigned to North America or to put it another way that is 40% for the rest of the world.

Doing the Research

Although the implementation is a somewhat recent experience the outlines of IPv6 have been around for some time. Like the original IPv4 development there are ideas and RFCs that may or may not be helpful. Like early IPv4 there is an element of experimentation in seeing what works in reality. It is therefore important that you do the basic research and reading around the subject to gain a full understanding of how IPv6 addressing works and how it is designed. My early reading of books around IPv6 produced a large amount of misleading and out of date information. It is still generally true that there are very few sources of information that give adequate hints and guidelines on how to plan your network. From personal experience "Migrating to IPv6" by Marc Blanchet (ISBN 978-0471-49892-6) is one of the better references on the subject.

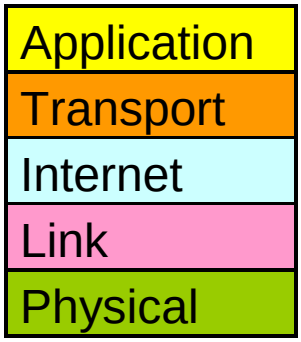
There is also quite a lot of information on the Microsoft web sites around IPv6 deployment and how it works under Windows. Microsoft, it has to be said, seem to have done a reasonably good job of rolling out IPv6, in particular under the Visa, Windows 7 and Windows Server 2008 (and later) generations of software where it comes enabled by default.

IPv6 is close enough to IPv4 to be familiar but far enough away from it to be easily mislead by what you already know about IPv4.

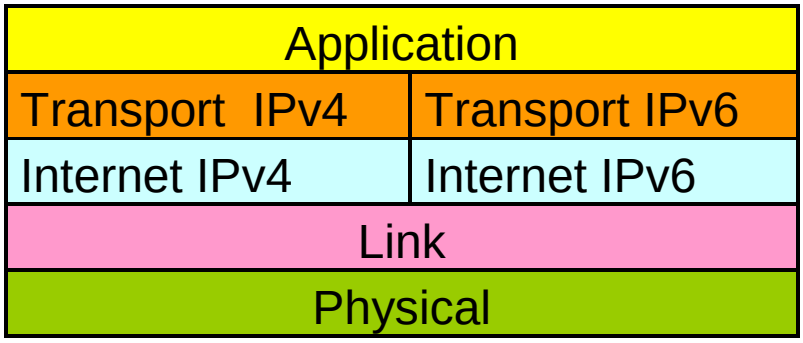
Dual Stack

Unless you wish to live in a particularly obscure form of isolation your network, in the short to medium term, is going to be dual stack. By this I mean that it will support IPv4 and IPv6. This is part of the design of how migration and deployment is supposed to happen. However, unlike other dual stack configurations (e.g. IPv4 and OSI say) the distinction between the stacks is not always as separated and as clear cut as one might assume. For instance it is perfectly reasonable for a DNS query over IPv4 to return a DNS answer which is an IPv6 address and vice versa.

So the standard model for the IPv4 stack which looks something like



becomes something more like



with the application layer now having to make the choice of protocol to use.

For systems with IPv6 enabled consideration needs to be taken of the fact that IPv6 becomes the preferred protocol. Given a DNS answer of both IPv4 and IPv6 addresses and with IPv6 enabled then IPv6 will be the preferred transport.

Connection preferences now look something like the following:

		Client Side		
		IPv4	Dual Stack	IPv6
Server Side	IPv4	4	4	X
	Dual Stack	4	6	6
	IPv6	X	6	6

It is also important to remember that deploying IPv6 on your computer (client or server) does not make your application dual stack or IPv6 compliant. An IPv6 connection requires that the IPv6 stack is enabled on your computer and also that the software that talks over that network is capable of handling the protocol. Because the IP address structures are different an application using an IP socket connection needs to be able to differentiate and store both of the structures. As a simple example an IPv6 client (e.g. Telnet session or database application) will not connect to a server where either the server stack itself is not IPv6 compliant or the server software (Tenet server session, database manager) has not been modified to accept IPv6 connections. The real issue will come for servers and applications that only offer IPv6 connectivity.

IPv6 Address Space

In terms of allocating addresses to your devices some thought needs to be given to how to plan your address space. Given the size of the address space in IPv6 there should be no shortage of space. Whilst the way that address spaces are handed out is down to your ISPs policy it is not unreasonable (and it is in fact suggested by the IETF) that an organisation of any size should be given a /48 prefix. This means, as a leaf node organisation of the Internet, that you have 65,535 subnets to play with and 64 bits worth of end node addresses.

What goes around comes around. In the good old days of the IPv4 Internet when the manuals told you to apply to your local registry to get a set of addresses for your network you could expect to get, at least, a class C address group. Each device in your network would therefore have a globally reachable address. Then along came NAT and firewalls which gave you more networks to play with, since you used non routable addresses behind the firewall but also gave you problems particularly if you had connections to other networks which were doing much the same thing – you had a clash of addresses.

IPv6 fixes all that. With such a large set of addresses available the intention is to go back to non-NAT, globally unique addresses. That's a definite maybe.

One issue to take into account is what happens if you change ISPs. With IPv6 your address space is controlled by your ISP and is allocated from their assigned range. Change your ISP and you change your assigned address range. Worse still is the fact that this arrangement is part of the way that IPv6 addresses the issue of large routing tables on the Internet core routers. They only need to know about ISPs assigned ranges so there is little chance of being able to have an independent static address range assigned to an end user network (unless you work for a company such as Google).

A second issue centres on what happens when you have more than one Internet connection for either resilience or capacity reasons. Each of these connections will have different address ranges assigned. Having multiple addresses is not an issue as far as the end node is concerned. IPv6 allows for multiple globally unique addresses to be assigned to an interface. There may be an issue with which address is used for an outbound connection but the major issues are around inbound data flows.

There are a number of solutions for data flows originated from outside an organisation – round robin DNS answers are just the most obvious. The problem for organisations, like ours, is more invidious where the bulk of the connections are originated internally to external servers such as web services. Policy based routing can direct the outbound flow over the preferred interface/network but inbound return flows will be controlled by the return address and the routing information held by the Internet routers. If source addresses are always from the same network prefix group then return flows will always be directed back along the network path defined by that network. The second network interface, when used for load balancing, will therefore not be used as desired.

By accident, if not design, IPv4 fixed this issue by using NAT and substituting the relevant interface address on outgoing packets so that the return flows came back to the same interface. The most obvious solution to this issue is to implement NAT at an IPv6

level in the same way as you did with IPv4. If that is your preferred solution then it may be worthwhile going the whole hog and not deploying globally unique addresses internally at all. Instead the IPv6 version of non-routable addresses is defined in RFC 1884 referenced as Unique Local Addresses. These are guaranteed to be “almost unique”.

For all the words written in the IPv6 documents around auto address allocation you will need some fixed points in your networks in the same way as and for the same reasons that IPv4 based servers and routers usually require a fixed address.

You And Your ISP

One of your first tasks should be to talk to your ISP and ask them for an IPv6 address range. This can be an enlightening experience (at least as of November 2010). We were lucky in that our corporate ISP said “certainly sir”. ISPs with a more domestic orientation have generally fallen into one of two reply categories. The first is “what’s IPv6” and the second is “we aren’t implementing that until we have to”. As a very general rule – the more corporate looking the ISP the more likely they are to have started down the IPv6 road. Even those with a track record have their blind spots; our current ISP had IPv6 on leased line services but not on ADSL circuits – a policy now updated to provide IPv6 over ADSL.

Picking Your First End Node

If you have a computer which is IPv6 enabled then you have already made the first step to trying out IPv6. If your routers/layer 3 switches have not been IPv6 enabled then IPv6 enabled machines (Windows Vista, 7, 2008 server and later) will, by default, have a link local address. A link local address is an IPv6 address that is only valid for the network segment to which it is connected. Since they are not guaranteed to be globally unique then you must specify the interface to which they refer. The interface specification depends on the OS you are working on but on Windows a ping for a link local address on the same network segment will look something like…

```
C:\WINDOWS>ping fe80::21f:29ff:fe42:1cb9%4

Pinging fe80::21f:29ff:fe42:1cb9%4 with 32 bytes of data:

Reply from fe80::21f:29ff:fe42:1cb9%4: time<1ms
Reply from fe80::21f:29ff:fe42:1cb9%4: time<1ms
Reply from fe80::21f:29ff:fe42:1cb9%4: time<1ms
Reply from fe80::21f:29ff:fe42:1cb9%4: time<1ms

Ping statistics for fe80::21f:29ff:fe42:1cb9%4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The above is a ping to the current machines link local address on interface 4.

Note that what you once knew as the ARP cache is now known as neighbourhood discovery under IPv6 although in many respects it is the same animal with different spots.

Once you enable global addresses on the LAN segment you are operating on a whole new ball game starts. Your computer will auto configure a globally valid IPv6 address (automatic address assignment, neighbour discovery) and will also work out where the routers are on that segment (router discovery).

A DNS server which returns an IPv6 result now (as well as an IPv4 result) will have the effect on your computer of preferring the IPv6 address to the IPv4 address since it no longer is bound by having only a link local address. If the hops between your computer and the node named in the DNS reply do not support IPv6 then the access will fail e.g. an IPv4 only firewall will discard the packets even if your link from your ISP is IPv6 enabled. At best this will result in a delayed response of many seconds whilst your web browser (or whatever) decides something is wrong and uses the returned IPv4 address instead. At worst the connection will fail. Our strategy has been to start at the furthest point and work inwards e.g. ISP to firewall, firewall to routers and switches, routers to end nodes on a per subnet basis.

Doing an Inventory

A simple inventory of your equipment and software is a good starting point. It will help you decide:

> How much of your existing kit is IPv6 capable (and if not how much is it going to cost to get to that state).
>
> Which software needs upgrading to make it IPv6 capable? This is potentially the more troublesome. As a example Oracle 11g is the first version that is IPv6 enabled. Changing database server versions is potentially much more traumatic than changing a network switch.
>
> Do you need to upgrade at all. E.g. you might decide that your Oracle database will only ever service internal users and that you cannot foresee a time when you will not be dual stack enabled.

Treat suppliers claims of IPv6 compatibility with some scepticism. On the basis of at least one suppliers definition of “IPv6 Ready” my old VAX 750 processor fits that description. If you don’t know what a VAX 750 is then it’s older than you are.

As a general rule if there is a function in IPv4 that you use then look for an equivalent in IPv6. Some examples: DNS, NTP, Firewall, switches (layer 3), routers, DHCP, DHCP relay, RIP, OSPF, EIGRP, SNMP, VLAN assignments. Layer 2 devices (unmanaged switches) do not need to be upgraded normally and switches without routing capability

e.g. VLAN management only should be OK provided that you don't want to manage them via an IPv6 address.

Don't forget the obvious things like printers, terminal servers, hand held devices. I have yet to see an IPv6 enabled VOIP phone but it is worth checking.

Certain devices may not require upgrading immediately. NTP servers should be OK for as long as you are dual stack (as far as I know time is the same in the IPv4 world as it is in the IPv6 world).

I have been lightly bitten by layer 3 switches which did not support DHCP relay (at least not in the way I expected). This means that it is not possible to discover the IPv6 addresses of DNS servers or DHCP servers. In turn this means that automatic node registration into the DNS server is not possible. In addition it means that should you have a laptop user who goes between IPv6 enabled sites then they will have to manually define their IPv6 DNS server addresses.

The Importance of DNS and DHCP

As you may have noticed there are a large number of characters in an IPv6 address. These combinations are not nearly as memorable as IPv4s simpler numbering and for this reason alone your DNS will have a bigger role to play. Combined with address auto configuration locating a node by other than name will not be something that most people will want to indulge in.

DHCP also becomes more complex. The "managed" and "other" flags, which are options that can be set on your router/switch as part of router discovery will define, for your VLAN, whether address allocation is "stateful" i.e. like IPv4 or stateless i.e. native IPv6 via router discovery. Note that specifying neither flag will still auto-create an IPv6 address. What the other flag does is allow information other than the address to be supplied by a DHCP server e.g. DNS server address, default domain, NTP server address etc.

Address reservations in DHCP have also changed and are no longer based purely on MAC addresses. Instead you will need to know the IAID and DUID values of your device (in Windows use IPCONFIG /ALL to see these values).

Beyond Your Local Subnet

Once you start to enable networks beyond your local subnet/VLAN the hiding places are harder to find. There is some good news however.

1. There are very few (as yet) DNS servers returning both IPv6 and IPv4 addresses for a DNS query. So even when you enable an IPv6 backbone and the link to your ISP then the chances of actually encountering an IPv6 address are quite slim. Andrews & Arnold are one specific site which is IPv6 enabled that I know about (www.aaisp.co.uk). My domestic connection uses an ADSL line from Goscomb (www.goscomb.net). Google

public DNS servers are fully dual stack – see https://developers.google.com/speed/public-dns/docs/using. Note that the http://www.thinkbroadband.com/isps.html site has a specific flag for ISPs providing IPv6 connections.

2. You still have control of your own internal DNS server so you have the option of either doing a slow rollout of DNS names with dual answers or you can opt (at least in the short term) to provide separate names for IPv6 addresses.

To NAT or Not to NAT

If you are looking for absolute answers to this question then walk away now. This is an area (like much of IPv6) which requires some consideration.

Talk to a network security specialist about IPv6 and tell him (and you will probably have to tell not assume he knows) that the intention is that NAT should not be used and he will probably tell you that your network will be compromised. But NAT is a side effect of not having enough IPv4 addresses available and not a security gain in itself. The primary security device is the firewall – all NAT does is hide the originating address mostly because it has to because it is a non-routable address. On the face of it then there is no reason why you should need NAT provided that your firewall is doing its job.

In fact there are good reasons to abandon NAT particularly if you are in the business of hosting large numbers of publicly available servers e.g. no need for port forwarding or husbanding addresses due to their scarcity.

There is an argument that knowing the real address of a device allows it to be targeted and for this reason the Microsoft auto configuration addresses do not, by default, conform to the generally suggested IEEE EUI-64 standard. Instead they add a degree of randomness to the address generation so that it does not directly derive from the interface MAC address. Losing the MAC address information in the IPv6 address, I have found, to be generally less than helpful. It can be turned off with

```
netsh interface ipv6 set privacy state=disabled store=active
netsh interface ipv6 set privacy state=disabled store=persistent
netsh interface ipv6 set global randomizeidentifiers=disabled
store=active
netsh interface ipv6 set global randomizeidentifiers=disabled
store=persistent
```

In many ways this is an area where custom and practice has taken on the cloak of being a natural physical law which cannot be broken.

There are some good reasons why NAT translation may be necessary even if it is not desirable.

1. To enable load balancing to multiple ISPs. As mentioned above inbound data flows are routed as a by-product of their destination address. Thus an outbound request may come back by another interface if there is only a single global address for the originator.

2. NAT enables you to change ISPs without having to renumber your internal network. There is a proposed standard for node renumbering to overcome this but I am not aware of it having been implemented in real life. This problem is partly addressed by DHCP address allocation but doesn't work for fixed addresses. The IETF RFC 6296 (http://tools.ietf.org/html/rfc6296) is a proposal for IPv6 NAT translation. At the time of writing this standard has not been ratified – a search for content on NAT66 will provide a plethora of information why NAT66 is a bad idea and also why it is currently the only real solution available. A number of firewall vendors have implemented this unratified standard. The alternatives involve a high degree of networking knowledge in obscure areas of Internet Protocol which may involve other unratified or unimplemented protocols.

   Note that the implication of NAT66 is that it is best implemented with the same size prefix internal and external to the firewall. If the aim is to avoid network renumbering then Unique Local Addresses should be considered for the internal network.

An interesting side effect of NAT under IPv6 for outbound connections is to consider how you know it is working. Under IPv4 an incorrectly translated non routable address would fail because ISPs would not route the source address. A valid global address under IPv6 is still valid and if it is not being translated will still work. A number of web sites exist that will tell you your external addresses – e.g. point a web browser at SixXS (http://www.sixxs.net/tools/ipv6calc/) or http://test-ipv6.com/ and this will tell you the IPv6 (only) address from which you are connecting. For a NAT translated address this should be the external network address.

There is some relief for larger organisation with many inbound connections. The network between the ISPs router and the organisations firewall needs to be unique (as it does for IPv4). Because of the plethora of available addresses this is likely to be a 64 bit prefix address which gives the opportunity to use 64 bits worth of node addresses which can be translated across the firewall to devices on the private side.

Email Considerations

In general SMTP mail works in a similar manner to IPv4. However several mail providers (including Google and Yahoo) have chosen to implement some anti-spam techniques as part of their IPv6 deployment. Assuming that you host your own email servers then you will need to make some changes to the DNS records for your organisation on your external Internet DNS servers.

Unless you choose to implement a similar set of checks then inbound IPv6 mail will just require that you add an additional MX record that references your mail server IPv6 address.

Outbound email will require that you declare AAA, PTR and TXT records as per the following example for your domain:

```
AAAA smtpout 2001:a88:1:e::f:1
PTR  smtpout 1.0.0.0.f.0.0.0.0.0.0.0.0.0.0.0.0.0.e.0.0.0.1.0.0.0.8.8.a.0.1.0.0.2.ip6.arpa
TXT  @       v=spf1 mx ip4:80.168.14.98 ip6:2001:a88:1:e::f:1 -all
```

The AAAA record nominates the address of the server.
The PTR record is the reverse lookup for the AAAA record
The TXT record implements the rules for Sender Policy Framework

Without these changes email to major providers will be rejected. More information on the SPF format is available online.

Contact Details

Savant Ltd
Dalton Hall Business Centre
Dalton Lane
Burton in Kendal
LA6 1BL
Phone: +44(0)1524 784400
Email: sales@savant.co.uk