IPv6 Case Study

Guy Edwards, Network Development and Support Officer, OUCS, Oxford University

The following is also published on

http://blogs.oucs.ox.ac.uk/networks/2010/07/15/the-state-of-the-ipv6-deployment/

At the ICT conference I gave a talk alongside Bob Franklin from Cambridge. Bob covered IPv6 in general and then I covered a deployment case study using our teams deployment plans but trying to make it as general as possible so that I'm describing

- a connection to the internet
- having services you run
- having customers you serve

…which should hopefully make it applicable to Oxford or Cambridge and any subunit.

This is a rough text version of the talk, I didn't speak from a script so it may differ slightly and in the talk I did skip a few minor ideas that Bob and I had accidentally overlapped on.

Note that you could make a start on these steps today, you do not need a IPv6 connection as your first step. A working connection is actually quite a late step in the preparation.

1) Perform a network device audit

In roughly 2008 we started an audit of all switch, router and firewall hardware on the backbone to record hardware and software versions.

The resulting list was then compared to what the vendor said the device could do. Some devices might have no IPv6 support, others might be partial, others might support it but purely in software as opposed to hardware. For a firewall supporting IPv6 in software might mean that the device can deal with multi Gb/sec rates of IPv4 traffic but only 80Mb/sec of IPv6 which is not workable.

When looking to replace items you may decide some devices might not need IPv6 support in their lifetime – it might be operating as a simple switch with no access controls nor management via IPv6.

Be prepared to test a sample device with the software version you are planning to run – as fairly new implementations IPv6 commands may have defects or simply fail to run. You might need to upgrade your device vendors operating system on the device or raise a support case with them. It's usual to upgrade the software for security issues however it's typical to stay on a release that it known to be stable so you may uncover the need to upgrade as you test your intended IPv6 deployment.

Rather than spending money to replace a large quantity of equipment, plan to replace devices during normal hardware maintenance cycles. That is, if a non IPv6 capable device is up for replacement in a year, then perhaps you can wait until this time to purchase new hardware that is ensure to support IPv6. Ask the vendor for a sample before you purchase many units and test it. Complain with technical detail if you find flaws while attempting your planned IPv6 commands.

For other equipment you may decide it's necessary to make a short term purchase in order to keep your deployment plans moving.

2) Perform an audit of services

The public services offered by our team were fairly easy to enumerate (DNS, NTP, SMTP etc), for each we recorded the software version and research the IPv6 capability. Where it was lacking the expected native IPv6 release version/date was recorded.

Slightly more tricky is enumerating the tools we offer that provide a service to our customers (in our case IT support staff), such as web interface that take an IPv4 address as the query in order to show DHCP reservations, or anywhere an IPv4 address is entered to register a device. In each case the tool needs to be rewritten to support IPv6 (which might be minor or major code changes depending on the tool) or needs to be planned to be replaced.

Then lastly behind the scenes we have all the scripts that ‘glue’ together the services; cron jobs that retrieve database data, scripts that update service configuration, backup and restoration scripts. We have roughly 150 of these, and an audit needs to be done to find places for instance with regular expressions looking for IPv4 addresses, replacing them with a regular expression library of some sort that can match either IPv4 or IPv6. I’d suggest using a regular expression library that can do this is better than having to write two separate scripts, or a script full of twice the logic decisions.

3) Build an IPv6 test network

We built a IPv6 only test network and deployed the same production services we run on IPv4 but entirely in the IPv6 test network

We documented any differences in configuration syntax and behaviour for setting up the service under IPv6

We also documented minor aspects, changes in utility names (e.g. ping6 instead of ping, ip6tables instead of iptables) since this step is also important in getting staff experienced in the configuration under IPv6 and also root out any surprises

Note that you **do not** need a functional IPv6 connection to any external network to undertake these steps.

4) Write a formal deployment plan

Can be peer reviewed, helps to straighten out all the minor aspects of deployment that might not have been considered.

The plan can then be shown to management and other teams to make them aware of the situation

We then approached a unit (e.g. customers) we knew to be technically able and constructive to work with (for a department this equivalent might be a research group or similar subset of users), asking if they would like earlier than normal access to an IPv6 connection supplied to their unit in return for feedback from them and under the understanding that this was not a production service – there might be some unforeseen consequences during development.

Having had a good experience with the first, we approached two other customers but I had made a mistake in our planning. We’d considered all the technical aspects of deployment but not the political aspects. The questions included:

- Can you prove that the university isn’t about to demand our IPv4 space back?
- Can you prove when the last IPv4 space will be given out in the university?
- Is the University making any money available to the units for the change?
- The supplied IPv6 connection has to be a fully production service with no issues

By this point we’d also joined the Cisco IPv6 deployment council so that we could give feedback on what IPv6 feature development we wanted from our current switch vendor. It’s under a Non Disclosure Agreement so I’ll not go further into this other than to say it’s technically valuable to us.

5) Produce a formal IP addressing Policy

It took longer than expected to produce the addressing policy, the main concerns being of making mistakes at this point in time that would be impossible to fix without severe disruption later and hence inflicted upon generations to come.

We used advice from technical sources including Southampton and Loughborough Universities as well as our switch vendor.

6) Where are we now?

- We’ve deployed a dedicated server to handle IPv6 traffic in and out of the university in place of passing it through the main university firewall, the later being capable of IPv6 in software only. When the next maintenance purchasing cycle is due this will be replaced with dedicated hardware.
- We’re aiming to supply IPv6 connectivity to our machine room in a controlled fashion (to avoid unexpected impact on other teams), once done we can start IPv6 enabling the core network services.
- The first unit will be supplied with an IPv6 connection as part of the initial trial once our security team is content that they are ready to handle dealing with security incidents on IPv6 based hosts (we’re expecting this to be probably within a couple of months).
- Our current university DNS/DHCP interface for IT support staff cannot handle IPv6 but the implementation of the replacement in the Oxford environment is a little complex. The replacement was hoped to be ready for mid August but this now looks impossible (this is a large project so I’ll do a separate post to cover this).