aarnet
Australia's Academic
and Research Network

AARNet's experience with IPv6

Glen Turner
Australian 2007 IPv6 Summit
2007-11-20

1. Motivation

The AARNet3 network has native support for IPv6.

Firstly, we wanted our customers an additional option for dealing with IPv4 address space exhaustion. NAT may not be a good fit to large universities: it requires per-connection state in the router, and thus is vulnerable to denial-of-service attacks; and requires the router to support every protocol used, yet universities are so large that even an accurate census of the protocols in use is laughable.

We wanted to make IPv6 available in time for considered adoption. We did not want a repeat of the Y2K vacillation leading to auditor-driven crisis and overspend on remediation.

Secondly, we did not wish to re-purchase expensive capital items such as backbone routers should a production demand for IPv6 appear. The simplest way to validate the equipment for IPv6 was to configure it; and once configured deployment was a small step.

Being ahead of customer demand was desirable. Universities are like ships, they take a long time to start turning but once in motion you do not want to be in their way. AARNet is a small organisation, even for an ISP. We wanted to roll out IPv6 through our network on our timetable, not that of our most demanding customer.

Thirdly, our users are at the cutting edge of computer networking, and supporting such research is one of the rationales for AARNet's continued existence.

2. The good

AARNet had already ascended the learning curve. We have offered IPv6 tunnels and conducted workshops since 2002. International academic computer networking conferences have discussed IPv6 for years. Other organisations may lack this deep background.

We used the same network design for IPv4 and for IPv6. The IPv4 topology was expressed using OSPFv2 and BGP. The IPv6 topology was expressed using OSPFv3 and BGP. The choice of stub areas, link metrics, and BGP import and export policies were the same for both address families. Despite these identical policies, IPv4 routing and IPv6 routing run as “ships in the night” with no interaction between them. The purpose of this design choice was to isolate the more important IPv4 forwarding from errors in IPv6 routing implementations.

2.1 Addressing

We used /64 addresses with ::1 and ::2 for router-router links. These links should not use EUI-64 addresses, since router addresses end up scattered through the network configuration. We do not want to track all of these references down should the router interface change (for repair or for network design reasons).

We used a /128 address for the router's control plane address (what IOS names “Loopback0” and JUNOS names “lo0”). This is analogous to the use of a IPv4 /32 address for the same purpose.

We used EUI-64 addresses on router interfaces to subnets containing hosts. Neighbour discovery allows hosts to

readily find the router's address. In retrospect a hard-coded /64 for the router's address may have been better and we may yet implement that.

For hosts we used stateless addressing, followed by DHCP6 to collect details such as DNS and NTP server addresses. Default routes are learned from Router Advertisments.

We use anycast addresses for DNS forwarders; again this is a reflection of our design for IPv4.

2.2 Interior routing

We chose OSPFv3 for interior routing. This has been solid on the Cisco and Juniper routers we use. The Cisco IOS syntax for OSPFv3 is different but better than the syntax for OSPFv2.

OSPFv3 uses IPSec to authenticate its traffic. On Juniper JUNOS this requires a IPSec configuration, which can be confusing as IPSec has so many permutations. On Cisco IOS configuring IPSec authentication is similar to configuring IPv4's MD5 authentication.

I recommend the use of OSPFv3 rather than using IPv6 extensions to EIGRP or RIPv2. Risk management of a deployment is much simpler when IPv6 routing is entirely independent of IPv4 routing.

This does not come without cost. Operationally, running two sets of routing protocols is more work than running one protocol. Making the IPv6 OSPFv3 topology the same as the IPv4 OSPFv2 topology is a near essential: doing otherwise radically increases the cost of running the second protocol.

2.3 Exterior routing

BGP can carry IPv4 routes over IPv4 or IPv6 neighbourings and IPv6 routes over IPv4 or IPv6 neighbourings. We chose to carry IPv4 routes over IPv4 neighbourings and IPv6 routes over IPv6 neighbourings.

We felt that the cost of doubling the number of neighbourings was well worth the advantages of total separation of the two address families; the ability to run IPv6-only peerings; and increased interoperability.

3. The bad

3.1 Resource consumption of two address families

Running two address families requires more resources than running once address family.

You can see from our configuration that we are running:

- *Two OSPF processes.* These track the same topology changes, and thus their CPU usage peaks simultaneously.
- *Two BGP neighbourings.* This doubles the number of BGP Hellos that need to be generated and checked. The BGP Hello processing is the load which limits the number of BGP connections to a router.
- *Two routing tables.* The additional IPv6 routes require additional memory. As the use of IPv6 grows and the number of routes will grow. Fortunately the falling price of dynamic RAM and the growth in routing table space needed for MPLS VPNs probably gives enough overhead in most routers. The RAM in many routers can be upgraded.
- *Two forwarding tables.* Forwarding table space is a limited resource. Some Cisco switch/routers implement it using a CAM table, Juniper routers implement it using static RAM. The forwarding capacity of a router cannot usually be upgraded without replacing an entire card.
- *Two sets of counters.* Hardware-based counters count packets and bytes flowing through interfaces. We want two sets of counters: once for IPv4 packets and one for IPv6 packets. The number and type of hardware counters in a router is very difficult to upgrade, as the counters are distributed across the interface cards.

Resource limitations can be difficult to spot.

For example, *Company A* routers had one set of hardware counters per interface. Those counters can generate

NetFlow for IPv4 packets or IPv6 packets but not for both address families.

Company B switch/router has a single CAM table for IPv4 and IPv6, with 25% of the CAM table allocated to IPv6. If every host has a IPv4 and a IPv6 address, then isn't that CAM table one quarter of the size needed? Should a hardware upgrade have accompanied the claim of IPv6 support?

Fortunately, MPLS VPN and VPLS have also caused demand for forwarding tables to grow much larger than the projected growth in the Internet routing table. But if you run VPN, VPLS and IPv6 you will need to check that the forwarding hardware is sufficient.

3.2 Less rich exterior topology

IPv4 networks have many interconnections and a great deal of effort is spent planning and tuning these links.

IPv6 networks have much less interconnection. The amount of traffic is low so a large effort is not justified. Running a congruent IPv4 and IPv6 exterior topology is stymied by the small number of ISPs which currently run IPv6.

3.3 Domain name system

EUI-64 addresses are a pain to maintain in DNS, yet are very nice in all other respects.

For servers we used stateless autoconfiguration and manually entered addresses into DNS.

For clients we currently do the same.

What we would like is for the address and router to be autoconfigured and then the host request a stateless DHCP. We would use this request to do a dynamic update of the DNS server. The new DHCPv6 server currently being developed by the Internet Software Consortium can be configured in this fashion.

This configuration requires servers and clients to be in differing DNS zones. That is desirable in any case, since we want to have servers in a DNSSEC-secured zone but want clients in an unsecured zone.[1]

3.4 Domain name system name resolution and black holes

When resolving names into IP addresses the IPv6 `AAAA` address should be tried before the IPv4 `A` address.

Hosts configured for IPv6 must be able to detect the lack of a IPv6 path to the host and fail back to the IPv4 path. This is usually done by the resolver library, which will only query for a `AAAA` if there is a global IPv6 address on at least one non-loopback interface.

Older IPv6 code, such as that in Windows Xp, does not do this. On those systems activating IPv6 whilst being on a IPv4-only network can cause large delays.

On recent systems it is still possible not to have IPv6 connectivity but to have IPv4 connectivity. This is usually because a IPv6 tunnel laid over the IPv4 infrastructure has broken. Depending upon the application a IPv4 connection may be tried after the IPv6 connection has timed out. However the user often gives up before IPv6 times out. Many Unix commands accept a `-4` parameter to force use of IPv4 in this circumstance. It is wise to configure the secondary mail exchanger with only an IPv4 address, as this lets mail through even if the IPv6 path is broken.

4. The ugly

4.1 Vendor box ticking

Every vendor seems to have “IPv6 support”.

But many router features are not available for IPv6.

For example, almost all routers have IPv6 traffic following the base topology

1 DNSSEC allows the address of every host in a secured DNS zone to be discovered. This leak of information does not matter for public-facing servers, as the same information can be found from a search engine; and there is large benefit in having a secured name to address mapping. For clients the leak of information allows targeted vulnerability scanning; and there is little benefit in having a secured DNS entry. Placing servers in a secured zone, say *example.edu.au*, and clients in an unsecured zone, say *client.example.edu.au*, is a reasonable security trade-off.

rather than MPLS-TE paths layered over that topology. If you are using MPLS-TE paths to implement an acceptable use policy then IPv6 traffic cannot participate in that policy.[2]

Another tactic is to handle IPv6 traffic in the CPU, which isn't called the "slow path" without reason.

4.2 Vendor software versions, feature sets and code trains

It used to be the case that IPv6 support was only present via unsupported patches. This has improved considerably.

For many products good IPv6 support is only available in recently-released code.

Some vendors have large feature sets, with differing "code trains" supporting differing feature sets. Check that IPv6 support exists in a "code train" with a feature set which suits your network. Beware of vendor claims of IPv6 feature support, often these will exist in a code train that you would not deploy.

4.3 Firewalls

In previous years firewall support for IPv6 was woeful. This has improved, but still not all features that are available for IPv4 are available for IPv6.

Firewalls often have a "allow unknown" default when IPv6 is activated by not configured. This is undesirable: as more devices ship with IPv6 enabled the firewall will grant these devices an unfiltered attachment to the Internet with no action by the network administrator.[3]

On host-based firewalls remember to allow all protocols not only across the IPv4 loopback network (127.0.0.0/8) but also across the IPv6 loopback address (::1/128). Otherwise traffic internal to the host which uses the loopback interface may behave oddly.

Some firewalls conduct Network Address Translation on all packets. These firewalls should be examined closely to ensure that the necessarily lesser range of supported protocols for IPv6 is adequate.

The wisdom of using NAT for IPv6 traffic should be questioned. Deep packet inspection is desirable: this carries much less state than address translation and it is always possible to discard deep packet inspection state with no effect on forwarding traffic. This makes denial of service attacks against the firewall much more difficult than the simple denial of service attacks which can disable a NATing firewall.

Many firewalls support OSPFv2. This is extremely useful when designing redundant firewalls in differing network cores. Few firewalls support OSPFv3.

Similarly to firewalls, many VPN devices do not support IPv6. You should ensure that these devices do not forward IPv6 traffic in plain text across the Internet rather than forward the traffic through the encrypted tunnel.

4.4 Middleboxes

There is a wide range of non-router middleboxes which munge traffic. These include packet shapers, load sharing devices, authentication devices, SSL end-points and the like. Almost without exception these specialist devices do not support IPv6.

4.5 Switch features

IPv6 support in routers is good. The same cannot be said of enterprise switches. These switches have a wide range of IPv4 features. The most basic of these are:

- *IGMP snooping* allows multicast traffic to be forwarded to only the ports which are a member of the multicast group, rather than being flooded to all ports.
- *DHCP snooping and source address validation* prevents IPv4 hosts from using addresses other than those supplied using DHCP.

2 MPLS-TE is often used to implement routing policies where particular customers cannot use particular classes of links. Academic and research networks often buy bandwidth cheaply on the condition that both parties are education or research institutions; communications with other parties cannot use those links. Defence contracts for IP VPNs often request traffic not cross links controlled by foreign-owned telecommunications providers.

3 For example, Hewlett-Packard network-attached printers will support IPv6 from 2007.

- *Quality of service class from DSCP* takes the QoS class from the IPv4 packet's Differentiated Services Code Point. This gives the benefit of endpoint-selected QoS to hosts without requiring hosts to run 802.1u priority marking or to run 802.1q VLANs.

There are many features beyond these, some “switches” even have firewall and VPN cards.

4.6 Validation

As can be seen, IPv6 is not yet at the stage where vendor claims and third-party certification can give reasonable assurance that equipment you purchase will adequately support IPv6 in your particular network.

This implies that the work needs to be done by the potential purchaser. Presently, only large networks validate their IPv4 equipment purchases.

Validation is a significant cost of deploying IPv6. More responsible vendor behaviour would reduce that cost and encourage IPv6 adoption, at the cost of less rosy statements from salesdroids.

4.7 Accounting and other back-end systems

Most ISPs run home-built accounting systems. These usually handle IPv4 addresses only. IPv4 addresses can be found in apparently-unrelated applications.

Updating these systems can happen either incrementally with application maintenance or in a rush when the ISP wishes to charge for IPv6. An incremental enhancement has less cost and risk.

Particularly problematic applications are:

- *Provisioning systems.* These systems write router and switch configuration fragments and will need to be updated for IPv6.
- *NetFlow collectors.* Cisco's NetFlow does not handle non-IPv4 traffic until NetFlow version 9. This is a complete re-design of NetFlow: it now uses templates to allow MPLS VPN and IPv6 traffic to have flow records. Unfortunately, most NetFlow collectors have not been enhanced to accept NetFlow v9.

  You should validate that vendor claims of NetFlow v9 support include the IPv6 template.
- *Network monitoring.* IPv4 and IPv6 are usually handled in distinct MIBs. A few network monitoring applications will not handle IPv6 addresses. Many network monitoring applications cannot communicate with IPv6-only hosts.

  Validate that a IPv6 MIB exists for each IPv4 MIB your current network monitoring uses. Validate that your network monitoring application can communicate using SNMP over IPv6.
- *Usage systems.* A system exists to convert network usage data (either from SNMP or NetFlow) into a charge against the customer's account. This application is intimately concerned with addresses and will require modification to support IPv6.
- *Billing systems.* Some customers will contest their charges. The billing system usually carries enough information to allow a customer service representative to check the reasonableness of a charge. This may require the billing system to support the IPv6 address textual format.

Applications programmers work to a much longer time-scale than network engineers. Enhancement requests for IPv6 support need to occur some years before the application is required to support IPv6. Systems analysts would do well to anticipate the requirement for IPv6 support.

Most software more intimately concerned with network engineering supports IPv6 or soon will.

- *Monitoring.* Nagios ✔ . HP OpenView ✔ , Tivoli NetView ✔ , CA Spectrum ✔ , BMC Patrol?
- *Element managers.* Cisco Works ✔ .
- *Capacity planning.* Torrus ✔ , Cacti ✔ , MRTG ✔ .
- *Intrusion detection.* Snort ✘ , will have support shipped in 2008.

- *NetFlow collectors.* Flow-tools ✗. Some newer projects support NetFlow v9 but their is not yet an obvious product to recommend.

Almost all network server software supports IPv6.

- *DNS.* BIND ✔.
- *Web server.* Apache ✔.
- *E-mail.* Sendmail ✔, PostFix ✔.
- *IMAP.* Dovecot ✔, Courier-IMAP ✔.
- *Time.* NTP ✔.

4.8 Proprietary protocols

Some of the protocols used in computer networks have no public specification. What we understand of those protocols is the result of many years close reading of documentation, examination of protocol traces, reverse engineering of programs and inspired guesswork. When those protocols are altered to support IPv6 our understanding of those protocols may be set back many years.

The consequences may be trivial, such as the lack of support for these modified packet formats in *Wireshark* complicating the investigation of network faults.

The consequences may be severe, such as a firewall no longer being able to inspect traffic using that protocol or an interoperable service losing that interoperability for IPv6. Reverse engineering is much more widespread than it appears.

In the years leading up to the migration to IPv6 reduce your risks by avoiding secret protocols where ever possible. Use SIP rather than Skinny; use OSPF rather than EIGRP; use IMAP rather than MAPI.

5. Strategies

Time is running out if you wish to use IPv6 as a potential remediation against the exhaustion of IPv4 addresses held by the Internet address registries.

In particular, equipment purchased today will need to run IPv6 within a few years. Add IPv6 support to the mandatory criteria for network equipment purchases.

For some product categories the claims of vendors cannot be relied upon. You will need to validate claims of support by testing of your particular network design.

Decide in advance how to handle non-compliance, since all vendors will fail to provide all the features they provide in IPv4 over IPv6. We decided that a workable level of current support, demonstrated continued development of that support, no obvious show-stoppers in the hardware design (such as IPv4-only ASICs), and a public commitment to IPv6 support equivalent to IPv4 support were the criteria to rate IPv6 support as meeting our “mandatory” evaluation criteria.

We try not to regress, although we recently failed when we installed a new web and mail server behind a IPv4-only firewall. This firewall is being replaced by a pair with IPv6 support.

We have a corporate policy of providing the same experience for IPv6-only users as for IPv4-only users. Exceptions require explicit management approval.

We try not to purchase from equipment categories known to be slow with IPv6 support unless the vendor can demonstrate support today. Without competition what will be the vendor's motivation to add that support once they already have your business? We avoided these categories where we could by altering our network design not to requirement them. For example, rather than use a middlebox appliance for load-sharing DNS forwarders we used OSPF to implement anycast.

Glen Turner
http://www.gdt.id.au/~gdt/presentations