The ISP Column
A monthly column on things Internet

July 2015
Geoff Huston

Revisiting Apple and IPv6

A few weeks ago I wrote about Apple's IPv6 announcements at the Apple Developers Conference (http://labs.apnic.net/?p=637). While I thought that in IPv6 terms Apple gets it, the story was not complete and there were a number of aspects of Apple's systems that were not quite there with IPv6. So I gave them a 7/10 for their IPv6 efforts.

Time to reassess that score in the light of a few recent posts from Apple.

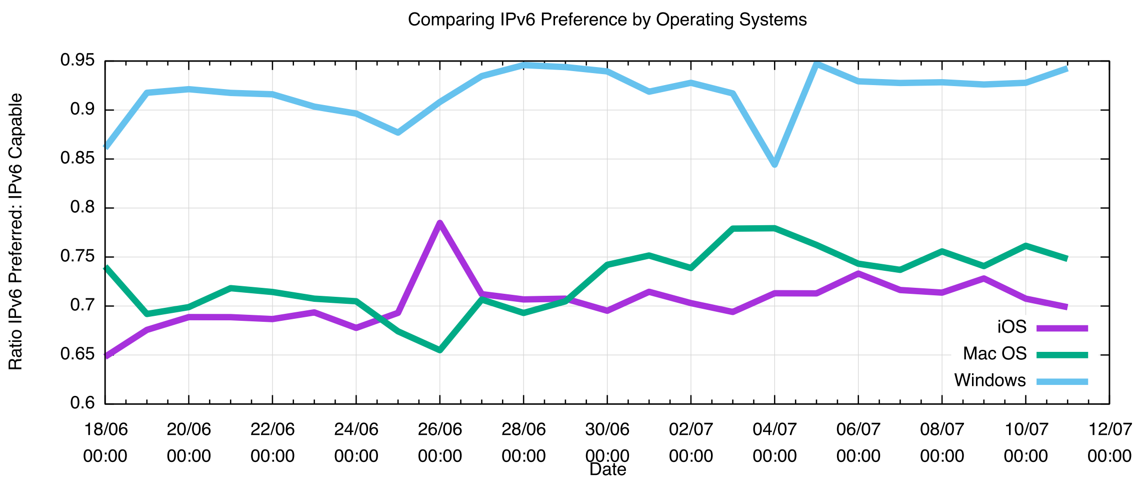
The first is a public note on the IETF's V6ops mailing list from David Schinazi of Apple (https://www.ietf.org/mail-archive/web/v6ops/current/msg22455.html) concerning contemplated changes to the so-called "Happy Eyeballs" behaviour of iOS and Mac OS X

Apple's current behaviour in a dual stack environment is that when there is a choice to use either IPv4 or IPv6 the Apple device will try to connect using what it thinks is the "fastest" protocol. The problem with this approach is that it generates some subtle forms of backward pressure in the IPv6 transition process. If devices have a bias to using IPv6 when possible then the more IPv6 is deployed then the less IPv4 is used. Deploying IPv6 in such an environment will reduce the intensity of pressure of use of address sharing mechanisms such as NATs, as the more IPv6 is used then the lower the level of use of IPv4. If the devices that have dual stack continue to use IPv4 then the use pressure on IPv4 NATs continue largely unabated in spite of an increasing IPv6 deployment. There is no implicit signalling through declining use of IPv4 of when IPv4 is no longer useful or necessary and the entire transition process may be prolonged for mare time than is strictly necessary.

Apple are contemplating changing their behaviour as of the forthcoming iOS 9 and OS X El Capitan releases. The browser's connection mechanism in a IPv6 equipped Apple device will generate DNS queries for the IPv6 and IPv4 address records, starting with the IPv6 query. If they get a response from the IPv6 address query first then the data connection will immediately commence with a TCP SYN packet sent over IPv6. If they get a response for the IPv4 address query first the system will wait for 25ms to see if there is an IPv6 address response to come. If so, then it will fire off an IPv6 connection, otherwise if the 25ms timer triggers without an IPv6 address response it will fire off an IPv4 connection. If the addresses are already in the host's local cache then it will pick the address and protocol with the lowest TCP round trip time recorded from previous TCP sessions. The essential change here is that it will weight the IPv6 RTT values by 25ms, biasing the host to select IPv6 in those cases where the numbers are within 25ms of each other.

Apple have observed that their tests lift the Dual Stack IPv6 preference to approximately 99%, which is a substantial change to the 50% IPv6 preference they have observed.

At APNIC Labs we have observed a slightly different behaviour (as shown in the following figure), where the observed preference rate for Apple systems in the Internet today is between 70% to 75%, slightly higher than the 50% reported by Apple. If the contemplated changes get to the 95% observed level of Windows then this would be a distinct change for the better.



It also appears that Apple includes support for IPv6 in the way in which its iOS Personal Hotspot operates (https://www.ietf.org/mail-archive/web/v6ops/current/msg22275.html). A conventional approach to implementing a personal hotspot in IPv4 is to use a Network Address Translation module in the hotspot domain, with the public-side NAT interface directed to the external interface. However this is one of those areas where you shouldn't simply substitute IPv6 in place of IPv4 and carry on. One of the underlying motivations in the design of IPv6 with its significantly expanded address space was to avoid the use of NATs in IPv6 altogether. If you can't NAT to create a personal hotspot in IPv6 then what options are left?

The answer to this question involves taking a piece of pre-NAT IPv4 functionality and using its current IPv6 equivalent. IPv4 address to MAC binding on local networks was originally provided by ARP. ARP required the host to send a broadcast message to all other IP hosts on the same network, and the ARP response would contain a response that bound a particular MAC address to a particular IP address. There was a way of binding together two distinct broadcast networks as a single logical IP subnet by using a technique called "Proxy ARP". A Proxy ARP relay passes ARP messages between subnets, substituting its own MAC address for MAC addresses that are relayed from one broadcast net to the other.

In IPv6 ARP's role has been largely replaced by the IPv6 Neighbour Discovery Protocol, which replaces the broadcast approach of ARP (together with some redirect functions of ICMP) with a richer set of messages that provide for router and neighbour solicitation and advertisements. The Neighbour Discover protocol is described in RFC4861. Like Proxy ARP there is an experimental specification of a Neighbour Discovery Proxy function where the proxy agent acts as a bridge between two distinct networks, obviating the need for distinct subnet addresses for the connected subnet. IPv6 Neighbour Discovery Proxies are described in RFC4389.

So if you want to avoid the use of NATs in IPv6 and you want to share an external IPv6 connection to a local subnet then it appears that the ND Proxy approach is a possible way forward. The clients of the hotspot believe that they are on a conventional IPv6 network, and the ND Proxy relays the ND messages to the external connected network.

So what are Apple doing here? The note from Apple has said that an iPhone with a dual stack cellular connection will support a dual stack client hotspot, and the approach being used here is the ND Proxy for IPv6. For those who want the implementation details then the next step is to consult the codebase for iOS, where the ND Proxy module is implemented. (https://opensource.apple.com/source/xnu/xnu-2782.1.97/bsd/netinet6/nd6_prproxy.c)

The Mac OSX systems, on the other hand, have no such support for ND Proxy. To quote from David's note to the V6ops IETF list, Mac OSX personal hotspot "does not support IPv6 because of the limited use cases, and the lack of demand for it."

And, finally, what about 464XLAT? Well this note from Apple says it simply: "we do not support 464XLAT."

A few weeks ago I gave Apple a score of 7/10 for their V6 story. The Happy Eyeballs change in their V6 selection policy is an important, and in my mind necessary, change. The ND Proxy support on iOS is also an important step forward, but I'm surprised (and a wee bit dismayed, I admit) that it has not been added to the MAC OS X code stream as well. And iPhones still don't support 464XLAT.

So where is Apple now with IPv6?

My score is now at 8.5/10.

Getting closer.

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 building the Internet within the Australian academic and research sector in the early 1990's. 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. He has worked as a an Internet researcher, as a ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer

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