Geoff Huston
July 2017

Notes from IETF 99 – The Other Bits

After pulling out the notes from the IEPG meeting and aspects of the DNS, here are the rest of the items that I personally found to be of interest at IETF 99 last week.

IPv6 Operations - Apple's Happier Eyeballs?

The way in which browsers leverage the potential opportunities offered by Dual Stack environment for detecting and using faster connections appears to be a moving target.

The original approach was to fire off two DNS queries, one for the A record (IPv4 address) and the other for the AAAA record. When both queries returned, and if the remote site had both A and AAAA records, then the application would initiate a connection using IPv6. If that failed then it would try in IPv4.

The problem is that "connection failure" is implemented in most TCP stack as a timeout after a series of connection attempts over an exponentially increasing backoff time. Windows takes 22 seconds to decide that a connection is unresponsive and return failure. FreeBSD, and Apple's systems use a 75 second timer, and Linux uses 108 seconds. Such timeouts may have had a role in the time of modems and a single protocol stack, where failure to connect had no 'Plan B', but waiting up to two minutes before switching to the other protocol is just not good enough, as it stretched users' patience way beyond any reasonable breaking point.

Back in July 2008 (yes, nine years ago) Stuart Cheshire described a process of launching concurrent connection attempts in both IPv4 and IPv6, with each process separately performing DNS resolution and opening a TCP connection (https://www.ietf.org/proceedings/72/slides/plenaryw-6.pdf). The first process to complete its connection (the one that receives the TCP SYN+ACK packet first) would assume the connection, and the other connection would be quietly closed.

But that's not what was written up as RFC 6555 as the so-called "Happy Eyeballs" approach. Some changes were made that were considered to be important to the success of the overall IPv6 transition. For this essentially unmanaged transition to work hosts, should prefer to use IPv6 when they can, and should avoid making extraneous connection attempts in IPv4. This would have the effect that the greater the deployment of IPv6 the lower the use of IPv4. For those running various forms of shared IPv4 NATs and CGNs this would presumably provide some additional incentive to deploy IPv6, as it would relieve the pressure on the CGN's IPv4 address pools. The "Happy Eyeballs" specification made two important changes: the DNS resolution part included a rendezvous point at its conclusion between the IPv4 and IPv6 processes, and the faster DNS resolution process waited for the other to complete before moving on., and, secondly, in the connection phase IPv6 is tried first, and if it has not completed in a reasonable time (300 ms is the common timer in this case) then an IPv4 connection is also started. The first one to complete the TCP handshake takes on the connection. The behaviour of this algorithm is consistent with the overall IPv6 transition objectives, where IPv6 is preferred, and IPv4 is only used when IPv6 is seriously lagging, and the greater the levels of IPv6 deployment the

lower the demand for IPv4, and, critically, the lower the levels of pressure on shared IPv4 NAT devices.

But Apple's Safari browser never implemented RFC 6555. MAC OSX keeps track of the round trip time estimate of all recently visited destinations in IPv4 and IPv6 (`$ nettop -n -m route`), and prefers the protocol with the lowest RTT estimate for a new connection. If the destination is not in this recently visited cache then it takes the average RTT of all IPv4 destinations in this cache and the same average RTT for all cached IPv6 destinations and use the lowest! In terms of the overall IPv6 transition this is somewhat questionable, and even in terms of selecting the fastest protocol it's not much better.

Apple is sharing its second attempt to grapple with these eyeballs. It uses Stuart Cheshire's original concept of essentially parallel connection processes, but if the IPv4 DNS phase completes before that IPv6 DNS, the IPv4 process will pause for 50 ms for the IPv6 DNS process to catch up. If the IPv6 DNS process completes in this period then the IPv6 TCP connection will be initiated. Otherwise the process moves to an address selection algorithm that orders the candidate addresses using recent RTT data, if available, to guide this choice. The underlying design trade off here is once of balancing speed with the larger objectives of the dual stack transition. Picking the fastest connection is one factor. Using IPv4 only as a last resort is the balancing factor, as this "last resort" approach is what will bring the transition to IPv6 to a natural conclusion as soon as it is viable. Personally, I'm still unconvinced that Apple has got this right, as 50ms is a somewhat unforgiving setting in terms of relative time differentials between the two protocols in many cases. The chance of picking IPv4 when the performance differential between the two protocols is marginal is still quite high in this scenario.

V6 Operations - To NAT or not to NAT

The saga of Unique Local Addresses in IPv6 continues.

The IPv6 address plan started out in much the same way as IPv4 some 30 years ago: every network could readily obtain its own unique network prefix so there was no need for local-use only addresses other than the host loopback addresses. However, in IPv6 this was not exactly true. Every network could readily obtain its own unique network prefix if they met the criteria for a network prefix allocation from the local Regional Internet Registry. As I recall, as a reaction to this situation the IETF created a new prefix, `fc00::/7`, defined as a pool of self-assignable prefixes that any site could use. These addresses were called "Unique Local Addresses", or ULAs, where the term "unique" was used as meaning "possibly unique!"

In many ways, this address pool resembles the RFC 1918 local address pools used by IPv4, which also are self-assignable with no guarantee of uniqueness. These networks are widely used in IPv4 deployments due to the prevalent use of NATs. But the concept of re-creating this in IPv6 attracts a strong reaction in the IETF. One quote from this discussion was "the presence of NPT6 or indeed any form of ULA-only deployment, save perhaps for ephemeral private networks is a bright line not to be crossed" At this point the use of ULAs is questionable is all bar completely isolated networks. It seems somewhat strange that over a decade ago when ULAs were an active IETF topic there was much in support for the concept, there is now effectively no interest in them at all!

Again this reminds me of the earlier reaction of the IETF to NATs in IPv4, where the strong negative response form the IETF at the time to undertaking any form of standards work on NATs made the problem far worse than it should've been. Not only are NATs subtle in the way in which they can affect applications, having each NAT vendor exercise unique (here I am using the term "unique in its more conventional manner!) creativity in the absence of a common specification meant that applications had a major problem on their hands.

I can only wonder why the IETF is choosing to repeat this rather poor decision. IPv6 NATs exist already, but once more their behaviours are not conformant to any particular standard. The resulting variation in behavours helps no one!

V6 Operations - IPv6 Multi-Homing is Hard

Over and above the expended address fields, IPv6 represented a set of very conservative incremental design choices over IPv4. However, in almost every case these choices have proved problematical. From fragmentation handling through to address architectures, it seems as if each of these changes created more issues than it had intended to solve.

One of these is multi-addressing. If an end-site is multi-homed with multiple providers, and it uses address prefixes from each of these providers, then the local host systems will pick up a network prefix from each of these providers. The challenge here is matching the local selection of a source address with the site's routing decision of which provider to use to reach a given destination. Sending the packet to the wrong egress will look a lot like source address spoofing, and the upstream provider may be using source address spoofing filters (BCP 38) and discard such packets.

Selecting the right egress gateway requires source address routing. This is a poor solution in many contexts, and other approaches seem to either be niche solutions or add further complexity to an already challenging issue. The latest of these responses is to perform a site-wide policy of selection of a single provider at any time and keep all hosts informed of this selection. This would allow hosts to pick the prefix of the operational provider and pass all outbound traffic to the associated gateway. It seems less than satisfactory from the customer's perspective to have paid for more than one uplink, yet allow only a single link to be used at any time. Work continues.

6MAN - Declaring IPv6 an Internet Standard

In the nomenclature of the IETF, publishing an RFC is just the start. If you want the specification to be described as a "standard" then you need to undertake a number of further steps. As described in RFC 2026, a full Internet Standard has to demonstrate technical excellence, prior implementation and testing, clear, concise, and easily understood documentation, openness and fairness, and timeliness. And if I could add one of my own, relevance.

Many RFCs are published at the entry level, Proposed Standard, and remain at that state. So common is this practice the IETF has these periodic discussions on proposals to remove the somewhat subtle distinctions between the various levels of standards documents and just call the entire collection "standards". But right now we still have these various distinctions, and in the case of IPv6 some folk perceive that the status of these documents as "draft standard" somehow gives an not-so-subtle signal that the specification is still incomplete and not ready for use.

Some would argue that this reflects the reality of IPv6, and the issues with dynamic configuration, the address architecture and fragmentation handling do require some further thought. Others would rather brush aside these outstanding technical operational issues and declare that the specification is done!

The process is now underway, and the 'core' specification, RFC2460 has been revised and published as RFC8200, an Internet Standard. The accompanying specification of Path MTU Discovery is published as an Internet Standard RFC8201.

But at this stage the other documents in the IPv6 specification are not exactly gathering a clear consensus to declare them done. One focus of comment has been the IPv6 address plan, and the division of the 128-bit address into a 64 bit network prefix and a 64 bit interface identifier. The rationale for this division is lost in a set of 20 year old debates, and to resurrect a very old conversation, it all involved a desire to use an addressing approach that provided some delineation between the use of

addresses for forwarding (location) and the use of addresses for endpoint identification. The compromise at the time was to try an approach called 8+8, where the 128 bit address use 8 bytes for location and 8 bytes for identity. The identity field was never all that satisfying. It became by default the IEEE 802 48-bit MAC address, padded out with a further 16 bits. The observation was made that if an endpoint preserved the same identity value when it roamed across access networks this became a privacy leak. So we then adopted an approach of putting a random value in these 64 bits and changing them periodically in order to blur any identification signal. But even so the question remains: why 64 bits? Why not 48? or any other value? One school of thought is that we should've learned a lesson from the Class-based IPv4 architecture and its subsequent transition to a more flexible architecture that allowed the boundary between the site prefix and the local identifier to be variably sized. Another school of thought says that a fixed size boundary in the address architecture is useful for interoperation, and allows tools such as site local auto-configuration (SLAAC) to operate as simply as possible.

The current working draft of the IPv6 Address Architecture document (https://tools.ietf.org/html/draft-ietf-6man-rfc4291bis-09) states "Interface Identifiers are 64 bit long except if the first three bits of the address are 000, or when the addresses are manually configured, or by exceptions defined in standards track documents." The manual configuration exception could be interpreted as saying "Interface Identifiers are 64 bit long, except when they are not!" It seems a rather furry statement for a supposedly definitive IPv6 Address Architecture Standard!

I must admit that I have little clear understanding of the nature of the distinction between the various categories of IETF standards RFCs, and the observation I would make is that so far we appear to have a large pool of IPv6-capable equipment that interoperates, so I'm not exactly sure about the exact value of this exercise to change the classification of these specs.

Homenet – A Most Complex House

Most home networks are extremely simple. Most home networks are simple flat Local Area Networks. Traditionally these were IPv4 broadcast networks with a gateway that is loaded with IPv4 NAT functions, a simple DNS forwarded and a DHCP agent. But the charter of Homenet sure does not see it this way. It looks at home networks as a far more complex regime with multiple security and access realms, multiple protocols and multiple policies. To quote from the charter: "by developing an architecture addressing this full scope of requirements: prefix configuration for routers, managing routing, name resolution, service discovery, and network security. The task of the group is to produce an architecture document that outlines how to construct home networks involving multiple routers and subnets."

This is no small ask. Already this working group has spun off work on a new routing protocol (Babel), a new name space (home.arpa), and the group is now in the areas of service discovery. I wonder if the goals are being set very high. For example the group is re-working DHCP to a home network equivalent HNCP, and is now deep into the consideration of securing HNCP with multiple keys and key sharing. Why?

There is also the question of multi-attached home networks. In the IPv6 context this immediately raises the ghost of the multi-6 work of a decade ago when the IETF commenced to grapple with the concept of how to attach end sites to the network with multiple service providers each offering the end site their own network prefix. At the time this work evolved to a NAT-in-the-stack approach called SHIM6. However, it never gained traction, and we seem to be back one more in Homenet to passing over the same space have much the same conversations again. Seems like old problems are new again. Who and how can individual components make decisions about which external prefix and which external service to use, and the conversation about whether such decisions are made for the entire site by the routers or for each host or even application by virtue of the address selection process used on the hosts or in the apps.

Often simple wins in the market and so far the flat single LAN dominates the home networking space. Unmanaged complex network structures are certainly risky and Homenet appears to be heading in a direction that heads directly into such complexity.

SIDRops

Thinking about IETF technologies that are struggling for deployment, the new working group, SIDRops met at this IETF. There are a number of issues with the current BGPSEC secure routing design setup that appear to be the cause of some levels of concern to network operators. Without the somewhat large scale investment in BGPSEC, and the strict requirement for connectivity adjacency, it appears that full BGP path validation is a very ambitious stretch goal. Indeed, so ambitious that it could be labelled as somewhat unrealistic at this point in time.

Without path validation, the origin validation offers only a thin veneer of route protection. An attacker only needs to reproduce the origin AS in an otherwise fake AS path to mount a routing attack that would not be readily detected by the origin validation mechanisms in the ROAs. This leads to the view that ROAS are effectively safeguards only against so-called fat-finger problems, where prefixes are inadvertently entered in the routing system by mistake. The level of protection offered by such an approach is not that different from conventional routing registries, but the cost of operating this additional mechanism is significantly larger. Little wonder that the reports of the level of production use of even origin validation in today's network show scant use, and the rather disturbing count of 10% of ROAs not actually reflecting the state of routing advertisements seems to attest some lack of interest in adopting this form of origin validation by network operators at the moment.

Even nerd types are fashion conscious, and one of the current fashions is attempting to match block-chain technologies to various operations. We've seen "namecoin" in the name registry realm, and in this SIDRops session we heard of a research study to see if a variant of bitcoin could be used as a form of address registry. It's not clear that this is a good fit, and I have to wonder if a better approach to address registry issues could be along the lines of published transaction logs to augment the current 'snapshot' format of registry publication. I'm sure that this topic will recur in coming meetings in various ways.

ACME

ACME is the automated certificate management framework used by the latest round of CAs, notably Lets Encrypt. This is a deliberate effort to remove the some of the barriers to traffic encryption, notably cost, complexity and arcane and convoluted procedures. Obtaining a domain name certificate for a domain name that is associated with a web site is as simple as being able to demonstrate control over the web site by being able to install a token on the site. A similar mechanism exists through being able to insert a token in the DNS.

But of course this is the IETF and you can never have too much of a Good Thing. So, where else can we apply ACME pixie dust?

One potential application is STIR, the effort to secure telephone numbers. The background here stems from the observation that the SS7 telephone signalling protocol is not secure, and when carriers installed bridges between SS7 and VOIP systems not only did this enable a new wave of IP-based voice calls, but it also enabled a new wave of robocall spam ware on the telephone system! The overall question being posed here is whether STIR could use ACME to generate certificates for "good" telephone endpoints? What proofs need to be built into the ACME service as some test or proof of elective control' of the number? Now one answer is just use ENUM - after all it was specifically constructed to represent the E.164 number system in the DNS and the idea here is that a delegation in ENUM is conditional upon assignment of the corresponding E.164 number block, so the ability to

place a token at the relevant point in the ENUM DNS would be equivalent to a proof of control. However, ENUM has its problem. Many operators do not either populate or delegate ENUM records, so the ENUM space is very sparsely populated.

If we are thinking about ACME for securing E.164 numbers, then what other applications could benefit from the same ACME magic? Why not use ACME for certificates for SMTP and IMAP, as doing it via HTTP and a dedicated web server can be cumbersome if the only reason while the web server has been set up is to support ACME interactions. Perhaps we could phrase the proof of control challenge in difference ways, such as with a SRV record in the DNS? Or should we avoid the DNS and do challenge/response in the target protocol where you are trying to introduce certification?

This form of pushing at the model to test the limits of its extensibility is typical of the IETF. Often it results in a parade of poorly thought out proposals, and sometimes truly bad ideas emerge, but sometimes it works. Sometimes it allows a different perspective to be applied to a problem space that enables new approaches to be deployed. ACME is going through this testing phase to see if we can extend the ACME model to offer simple and effective encryption services to a variety of applications and services.

Much more happened at the 99th IETF in Prague of course. The slides and materials for the meeting can be found at https://www.ietf.org/meeting/99/

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net

Disclaimer

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