The ISP Column
A monthly column on things Internet

October 2019
Geoff Huston

DNS Wars

NANOG is now quite an institution in the Internet, particularly in the North American Internet community. It was an offshoot of the Regional Techs meetings, which were part of the NSFNET framework of the late 80s and early 90s. NANOG has thrived since then and is certainly one of the major network operational forums in today's Internet, if not the preeminent forum for network operators for the entire Internet.

The 77th NANOG meeting was held in Austin, Texas at the end of October and they invited Farsight's Paul Vixie to deliver a keynote presentation. These are my thoughts in response to his presentation, and they are my interpretation of Paul's talk and more than a few of my opinions thrown in for good measure! (https://pc.nanog.org/static/published/meetings/NANOG77/2033/20191028_Vixie_Keynote_Dns_Wars__v1.pdf).

The DNS

The DNS is a fundamental part of the infrastructure of the Internet. Along with IP addresses, the namespace was considered to be the shared glue that essentially defined the Internet as a single cohesive network. Oddly enough, the issues with address exhaustion in IPv4 created a schism in the address framework that led to the adoption of a client/server architecture for the Internet that increased the reliance on the namespace as the consistent common framework for the Internet.

We refer to the DNS interchangeably to refer to a number of quite distinct concepts. It's a structured namespace, a distributed database, the protocol we use to query this database and the servers and services we use to make it all work. Little wonder that wherever you look on the Internet you will find the DNS.

Paul started his presentation with a description of the 1648 Peace of Westphalia, which is a rather odd place to start recounting the evolution of the DNS. This was a diplomatic congress between sovereign states. The principle of sovereign rights was established though this arrangement and the modern definition of a *nation* determined which parties had a seat at this particular table. A *nation* was defined as a geographic territory with defended intact borders, a principle that one could characterise as recognition of the rule of the strongest within bounded domains. This physical definition of a nation and the associated concept of *national sovereignty* has carried forward to the national structures of today's world order. The realm of sovereignty encompassed land and sea, and subsequently expanded to air, space (or at least the bits of it close to the earth) and now some aspects of the realm defined by information technology.

Let's park that concept and return to the evolution of the DNS. The original model of the name system for the networks of the 1980s was a way for a computer to conveniently name other computers connected to the same network. The initial method still exists in most systems as *hosts.txt* which is a simple list of names and the corresponding protocol address of the named computer.

The initial distribution of names was via flooding of a common copy of the *hosts file*. Pretty obviously this does not scale, and the frustrations with this naming model drove much of the design of the DNS. The DNS is a hierarchal name structure, where every nodal point in the namespace can also be a delegation point. A delegation is completely autonomous, in that an entity who is delegated control of a nodal point in the namespace can populate it without reference to any other delegated operator of any other nodal point. The implementation of the matching namespace as a database follows the same structure, in that an authoritative server is responsible for answering all queries that relate to this nodal point in the database. Client systems that query these authoritative services also use a form of hierarchy, but for somewhat different reasons. End systems are usually equipped with a *stub resolver* service that can be queried by applications. They typically pass all queries to a *recursive resolver*. The recursive resolver takes on the role of traversing the database structure, resolving names by exposing the delegation points and discovering the authoritative servers for each of these zone delegations. It does so by using the same DNS protocol query and response mechanism as it uses once it finds the terminal zone that can provide the desired answer.

There are perhaps three reasons why we believed that this was a viable approach.

- The first is that all resolvers cache the answers, and all answers come with a suggested cache time. Extensive caching within the resolver system suppresses queries.

- The second is that we used an exceptionally lightweight protocol that used stateless anonymous queries. UDP is exceptionally efficient, and the deliberate excision of any details of who was asking or why was intended to ensure that the answers were not customisable. This meant that the cached answers could be used to reply to future queries without the risk of breaking some implicit context associated with an answer.

- The third reason is that we aligned the resolvers with the service infrastructure. Your local ISP operated the DNS recursive resolver. This meant that the resolver's cache was nearby, and near has a much better chance of being faster than remote.

Ever since then we've been testing out these reasons and discovering how we can break these assumptions!

Episode 1 – The Root Wars

It appears that the first DNS War was fought over the transition of DNS names in `.com`, `.net` and `.org` from a USG contracted service that was offered without cost to customers to a charged service. To some the transition of the DNS into a commercial monopoly raised some issues. Why was one entity allowed to reap the considerable financial rewards of the now booming DNS while all potential commercial competition was locked out?

Various efforts were made in the mid-90s to compete with the monopoly incumbent operator, Network Solutions, by standing up alternate root servers that contained more top-level domains. The most prominent of these was an effort called AlterNIC, but it was not alone. When one of AlterNIC's founders hijacked the InterNIC website for three days in 1997 it led to civil lawsuits followed by a US Federal wire fraud prosecution.

The pressure for competition in the DNS did not go away, but the path through alternate root servers slowly faded out. The alternate path, namely competition in name registration services and the controlled release of additional names in the root zone, was the path that was ultimately followed. It is still debated today as to whether these moves achieved their intended objectives of enhancing competition in the namespace without adding confusion and entropy to the DNS, and the somewhat robotic continuation of expanding the root zone by ICANN appears more and more nonsensical as the years roll on.

Episode 2 – Sitefinder and Zone Contents

The next episode of the DNS Wars was the Network Solution's Sitefinder debacle. Network Solutions administered the *.com* zone as a registry operator. They added delegation entries to this zone according to orders passed to them by registrars. Search was gaining in popularity (and in intrinsic value) and it was noted that users were often confusing search terms with domain names.

Network Solutions decided to exploit this by synthesising a wildcard in the domain, effectively directing all queries for names that did not exist in .com to a search engine rather than conforming with DNS standards and responding with NXDOMAIN. After some drama and much legal posturing, the wildcard was withdrawn.

It's hard to tell now if the outrage at the time was about the seizure of as-yet undelegated domain names or this implicit seizure of search. In retrospect, the latter was the more valuable heist.

But the NXDOMAIN substitution issues did not go away.

Episode 3 – Open Resolver Wars

The browser vendors had decided that a single input element would be used for both search terms and searches. The result was a substantial cross leakage of search terms and DNS queries. Search engines gained valuable insights into popular but as yet undelegated domain names, an asset that was previously the exclusive property of DNS registry operators and DNS operators could capture a search session by substituting a search engine pointer in place of an NXDOMAIN query.

The emerging monopoly of Google search was not exactly uncontested, and when Google managed to obtain a default position in some heavily used browsers there was a reaction to try and redirect users to an alternative search engine. The DNS was co-opted in this effort and OpenDNS tried to achieve this with a recursive resolver that performed NXDOMAIN redirection into a search engine, in a reprise of Sitefinder. For a short period, OpenDNS also redirected the domain name `www.google.com` to a different search engine. Within a few weeks, Google launched their public DNS on quad 8 and based the service on absolute integrity of both positive and negative responses in the DNS. A 'trustable' DNS tat undertook to never lie.

Oddly enough the result is that Google's public DNS offering is now totally dominant in the open resolver space. If this was a three-way struggle between infrastructure-based DNS, Open Resolvers and Google's Open Resolvers, then it looks like Google won that round.

Episode 4 – Client Subnet Wars

The next episode of DNS struggles has been the Client Subnet wars. The deliberate excision of any details of who was asking or why was deliberately subverted by attaching a Client Subnet record.

This privacy-destroying initiative was largely due to Akamai. For reasons best known only to Akamai, this particular content distribution network was not a keen fan of using the routing system to steer the client to the closest instance of a replicated server set through anycast. For them, anycast was seen as suboptimal. Instead, they used the DNS. On the assumption that every client used an infrastructure-based DNS resolver, then the location of the resolver that they used for queries and their own location were close enough as to be treated as the same location. When a query came to Akamai's DNS servers the source IP address was used to calculate the client's location and the response was generated that pointed to the Akamai server set that was calculated to be closest to the user. No hand-offs, no additional round trip times, no more overhead. And if the recursive resolver cached the Akamai response then so much the better, as all clients of this recursive resolver would obtain the same response from the Akamai DNS servers in any case.

Open Recursive Resolvers are not necessarily located close to their clients. The result was that Akamai's location-derived response was at times wildly inaccurate and the Akamai content service was abysmally slow because a remote server was being used.

It seems odd in retrospect, as there are many ways that this could be solved, but the mechanism that was bought to the IETF for standardisation was to use the extension mechanism for DNS, EDNS(0), and record the subnet of the client into this field. Recursive resolvers were meant to perform a local cache lookup on the combination of the query name and the client subnet, and a cache miss required a query to the authoritative server with the client subnet information still attached to the query.

There are a whole set of reasons why this is a completely insane approach. It destroys DNS privacy, in that authoritative servers are now aware of the identity of the end client. The notion of what is a "subnet" and what is a client address is evidently too hard a concept to grasp for some implementors and the full client IP address is seen all too often in the ECS field of the query. The CDN does not provide the recursive resolvers with a map of their servers' locations so that the recursive resolver and optimise its local cache, as that of course would be an unacceptable leak of the CDN's privacy, but of course in the warped world of CDNs its quite acceptable to undermine the individual user's privacy, as that just doesn't matter. The local recursive resolver cache is now under pressure, as it now has to add the client subnet as a lookup key into the local cache, so local caching becomes less efficient. Also, there is the consideration that if the server realises that the client is poorly served it is perfectly capable of redirecting the client to a closer server. Any delay in the steerage function is likely to be more than compensated by the benefits of using this closer server.

It's hard to see Client Subnet as an optimisation, and far easier to interpret this technology as a deliberate effort to pervert privacy in the DNS and deploy the DNS as one more tool in the ongoing effort to improve mechanisms of user surveillance and increase the efficiency of monetising Internet users.

Episode 5 – Today's DoH/DoT Wars

And now we have the DNS over something Wars. Without a doubt, this is now a complex issue, and the motivations of the actors are sometimes not easy to discern. At its heart is the observation that almost every Internet transaction starts with a DNS lookup, and if I were able to observe all your DNS queries as they took place, then I would probably be in a position to assemble a comprehensive up to date profile of you and your activities. In terms of surveillance data, the DNS can be seen as the data motherlode. The Snowden material showed that such data is not just of commercial interest, but also a topic of keen interest to state actors. The IETF embarked on a DNS privacy path. If this wasn't enough, the DNS is now the control point for many if not most cybersecurity functions.

Pushing the recursive resolver deeper into the network means that the DNS conversation between the client stub resolver and the recursive resolver may transit a far longer path across the network, and that lengthened path opens up an unencrypted query and response to a larger set of actors who could inspect, or possibly alter, the DNS transaction. The Snowden papers described some NSA activity along these lines.

The first outcome of the DNS Privacy Working Group was the definition of stub-to-resolver encryption, using TLS. The IETF decided to use TCP port 853 for this method, allowing the port 53 port number to remain as DNS (unencrypted) over TCP. The TLS setup may look like a heavy price to pay, but when you consider that a stub resolver will normally keep a single session open with a recursive for an extended period and there is TCP Fast Open to allow fast session re-establishment, this starts to look pretty much the same as DNS over UDP in terms of performance, and the encryption secures the stub-to-recursive conversation against observation and interception.

Then came the specification of DNS over HTTPS. It's not a new idea, and there are xxx over HTTPS implementations for many values of xxx, including IP itself! But there is a difference between hacking away at the code and standardising the approach. HTTPS is commonly seen as the new substrate of the Internet as it passes through firewalls relatively easily, the content can be readily masked in opaque padding and jitter generators, the combination of TLS 1.3 and encrypted SNI (ESNI) really does hide most of the meaningful visible parts of any session, and it resists middleware inspection and alteration. Many vital functions and services use port 443 so it is simply not an option to block this port completely. Why prefer DoH over DoT? DoH achieves everything that DoT delivers, but also embeds itself in all other traffic in a manner that can make it all but impossible to detect. This is about content hostels, where a single IP address is used by thousands of different content domains, and combine that with ESNI in TLS 1.3, where the distinguishing name is not shared in the clear then it is pretty clear that DoH can be used in a manner that evades most common (and cheap) forms of middleware detection and potentially some of the even more expensive detectors as well.

Even so, is this level of security going to be enough in any case? To put it another way, there is a theory that if the DNS is too complex for the Chinese Communist Party then they will stop filtering the DNS. There is also a theory that this is complete nonsense!

But if you are motivated enough to hide in the packet crowd, why not run an entire VPN session over port 443 with TLS 1.3 and ESNI? Hiding just your DNS queries is not enough if you want to conceal the entirety of your network activity and constructing a secure environment from a distinct and separate set of tools is often far less secure than the more comprehensive approach offered by a modern VPN with current TLS behaviours.

So why DoH at all? It doesn't appear to be solving a technology or a performance issue that is not already competently addressed in DoT. But there are compelling drivers behind DoH and they appear in the commercial landscape of today's Internet. The major issue is the tensions between applications and everyone else! If much of the value of services in the Internet is based on the knowledge of the end user's behaviours and preferences, then applications are hardly motivated to share their user-driven activity with anyone else. Using the platform's stub resolver is a leak, using the service provider's recursive resolver is a leak, and using transmissions in the clear is obviously a leak. If an application wants to limit the extent of information publication to itself and its mothership then it needs to avoid common infrastructure and drive itself through the network using secure channels. DoH can do this readily. And where all this is played out is in the world of mobile devices, where the value of the market sector and the services and transactions that occur in this sector dominate all others. Today's networks act as both a data collection field and platform for the delivery of data-steered ads. Everything else is incidental.

Episode 6 – Resolverless DNS Wars

Not only is the DNS used extensively in this manner, but also the web community has been energised to bypass these mechanisms as a new measure, and now we are contemplating a future network that features "resolverless" DNS. Like server push for HTML, resolverless DNS can make the DNS faster by preloading the resolution outcomes before the application may need to use them.

Currently, this is the topic of an IRTF research group item where the content itself can push DNS outcomes, but the pragmatic observation is that there are few impediments to this approach in the browser world. Push is already well established as a means of improving the time to load for content and there is little difference in pushing style sheets, content, scripts and DNS resolution outcomes.

Interestingly, a protected session, such as TLS, is considered to be good enough for push and DNSSEC validation of the pushed content is not considered necessary by resolverless DNS proponents. This strikes me as irresponsibly naive, and if content can push content then the recipient should be absolutely required to validate the veracity of the pushed data.

But perhaps the position makes more sense if you view this as a major divorce, where the web is separating itself from the Internet and wanting to sever all forms of inter-dependence with the rest of the Internet. Why share any of that user data when you can keep it all? So, when we talk about applications ingesting Internet infrastructure functions into their own space perhaps, we are not really talking about applications in a generic sense, but instead are focussing entirely on the web platform, browsers and their ecosystem of HTML-based applications.

It's challenging to predict how this will work through, but perhaps there is already one emergent factor that we need to consider, and that is the Peace of Westphalia and the concept of a nation being defined by its adequately defended borders.

The IT Corporate Nation State

In a world where one corporate entity provides the operating system for some 90% of all handheld computers (Google with Android), share the same corporate entity is used as the browser by more than 70% of users (Google with Chrome) and where a single open resolver service is used as the preferred resolver by some 10% of users (Google with Quad 8), then if the Internet was regarded as a distinct realm of human activity on a peer level with realms defined by land and sea, then Google's ability to assert sovereign rights over huge swathes of the information technology space, based on the ability to defend its assets, must be admitted. By that reasoning, the Westphalian model of nation-states applies here as well, and regulation is necessarily replaced with negation.

When the Mozilla Foundation announced its intention to ship the next version of its Firefox browser with a default setting that both enabled DoH and directed DoH to Cloudflare's Open Recursive Resolver service, the United Kingdom called for a "summit meeting" with Mozilla. This was not the enacting of legislation, the adoption of a regulation or any other measure that is conventionally available to a nation-state, but a meeting of a different nature. Is this the resurgence of quasi nation-states such as the Honourable East India Company, a joint-stock company that ran its own army (twice the size of the British Army at its peak in 1803), fought its wars and established and defended its borders in a manner that was fully consistent with the actions of any other nation-state?

Part of the new world order is that the space defined by the actions of applications is well beyond the traditional domain of communications regulation and even beyond the domain of regulation trade and commerce. Applications use communications as a service, but they do not define it. This is a new space, and the sovereign rights of nations are finding it extremely challenging to assert that they have primacy when they cannot defend their borders and cannot unilaterally enforce their will. Is the new definition of information technology nationhood equating to the ability to impose the national will on end-users irrespective of physical land and sea borders?

The Internet rode a wave of the deregulation of telecommunications. What deregulation meant was that enterprises were no longer confined to offer a standard service at a highly regulated price. Deregulation meant that companies were driven by user expenditure and user preference, and accordingly user preference became the subject of intense scrutiny. But, like the supermarket retail industry, knowing what the customer prefers is one thing, but knowing how customer preferences are shaped and influenced is an entirely different realm, because such intense scrutiny and acquired knowledge allows the enterprise to both shape preferences and then meet them. The Internet has been changed irrevocably from being a tool that allows computers to communicate to a tool that allows enterprises to deploy tools that are intended to monetise users in a highly efficient and effective manner.

We have reached a somewhat sad moment when it is clear that the DNS has been entirely co-opted into this regime. Sadder still to think that if this is a new realm of national sovereignty then our existing nation-state world order is just simply not able to engage with the new IT corporate nation-states in any manner that can curb their overarching power to defend their chosen borders. The 1648 Peace of Westphalia has much to teach us, and not all of the lesson is pleasant.

I have to thank Paul Vixie for his NANOG talk in looking at the evolution of the DNS and the Internet through this particular lens.

Disclaimer

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

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