**The ISP Column**
*A column on various things Internet*

October 2021
Geoff Huston

# On DNS Openness

When we deregulated the telephone industry, we replaced these national monopolies and their vertically bundled structures with a collection of separate enterprises whose actions are orchestrated by market forces rather than by the dictates of the incumbent monopoly telco. This was a comprehensive upheaval to the telecommunications industry, and one aspect of this broad sweep of changes was in the role of the regulator. Previously it was a rule-based framework: Is the incumbent playing by the rules we imposed on them? But now that this entire space is now just a collection of markets the regulatory role necessarily changes. The regulatory question becomes: Is the market operating *efficiently*? Or we could use a slight variant of that question and ask: How *open* is the market?

Normally this openness question is a question about barriers to competitive entry for new providers into the market. It's a question about the extent to which incumbents have been able to distort the market to benefit themselves by shutting out potential competitors and ultimately as a cost imposed on the consumer. However, there is more to this question about the use of markets as a signalling mechanism across a diverse collection of intertwined producers and consumers. How effective is the market as a signalling mechanism across these entities? Is the market providing clear signals that allows orchestration of activity to support a coherent and robust service? Is the market-driven evolution of the delivered product or service one that is chaotic and periodically disrupted, or are such signals of change clearly broadcast and smoothly adopted?

If you are a regulator with a charter of market oversight you might want to look at the telecommunications market through this lens of assessing its openness. And that's pretty much what happens with BEREC, the Body of European Regulators for Electronic Communications (https://berec.europa.eu/). BEREC held a workshop at the end of September on the topic of "DNS Openness". I was invited to present on this topic, and I'd like to share my thoughts on this question in this article.

> The presentation I used at the workshop can be found at https://www.potaroo.net/presentations/2021-09-27-berec.pdf.

## A "Consistent" DNS?

I should qualify BEREC's perspective of this question at the outset, as it is a subtle but important qualification: To quote from the brief I was given for my contribution to this workshop: "When considering the openness of the DNS, BEREC is not specifically referring to openness in a competitive sense, but rather in the sense that of whether users access and distribute information and content, use and provide applications and services, and use terminal equipment of their choice, irrespective of the user's or provider's location or the location, origin or destination of the information, content, application or service?" This re-phrasing what was intended by the term *openness* introduces the issue of DNS consistency and uniformity.

In so many ways this question of openness and consistency seems to be completely redundant when we are considering the DNS. If BEREC's question about DNS openness is a question about the ability to access the DNS and consistency of DNS responses across all users of DNS services, then that question is answered by observing that the primary objective of the DNS has been to behave in exactly this manner.

The DNS was designed to deliver the same answer to the same query irrespective of the circumstances of the querier. It leverages this property of uniformity of responses to make intensive use of caching so that DNS information is dispersed across DNS servers in order to serve DNS information far faster than a conventional end-to-end query/response protocol would otherwise permit.

DNS servers are intended to be *promiscuous* responders. The servers' DNS response is generated without reference to who is asking, the supposed location of the client, and the DNS platform the client is using. DNS servers will respond to all queries. The content of the response is not supposed to depend on the origin of the information used to form the response, nor the specific DNS platform used to host the DNS server, nor the location of these servers.

This means that every Internet user who is connected to the Internet can access the information that is in the DNS, and indeed that access essentially defines what is the Internet. The DNS is the Internet, not a selectively accessible attribute of the Internet.

But the reality of the DNS is a little tawdrier than this high-minded ideal.

Does everyone see the same DNS? Are we all part of the same Internet?

The answers to those questions today are respectively "No!", and "I really don't know any more, but I sure hope so!"

## A "Deliberately Inconsistent" DNS?

National regulatory frameworks impose a constraint on domestic service providers ideas to present the "correct" DNS resolution of certain DNS names. This is very common in many regimes and is not only part of the highly commented "Great Firewall" of China, but various DNS requirements exist in the United Kingdom, the United States, Australia, India, Russia, Iran, Vietnam, France and Turkey, to name just a few. In some cases, the local measures prevent a DNS name from being resolved at all, others synthesis a "no such domain" (NXDOMAIN) response while in other cases an incorrect response is synthesised (https://www.potaroo.net/presentations/2013-08-29-facebook.pdf).

As an Australian I am more familiar with the Australian legislative framework than that used in other regimes. In Australia Section 313 of the Telecommunications Act of 1997 (https://bit.ly/3B8tOjp) describes legal obligation of carriers and carriage service providers to "do [their] best to prevent telecommunications networks and facilities from being used in, or in relation to, the commission of offences against the laws of the Commonwealth or of the States and Territories."

In practice, the application of these provisions has resulted in the deployment of mandatory-to-implement DNS blocks by Australia ISPs on DNS names associated with proscribed offensive material, material relating to violence, terrorism and various forms of criminal behaviour.

It has had its ludicrous moments, such as in 2006 when Australian publisher Richard Neville had a website taken down without recourse because it had a fictional transcript of the Prime Minister of the day apologising to Australians for Australian involvement in the Iraq war. It has also been captured by lobbyists for the copyright protection industry and various content services, such as The Pirate Bay and IsoHunt, have had their DNS names added to the supposedly secret Australian DNS block list.

There are other cases of local alterations to DNS answers. Some of these have been in response to the national DNS blocking measures. A commonly cited example is in Turkey from 2014, where it was reported that many local users turned to use Google's Public DNS service in response to government measures to block Twitter (Figure 1). The countermove was to direct Turkish ISPs to re-route the IP address prefix 8.8.8.8, used by Google's DNS service, to their local DNS resolver.

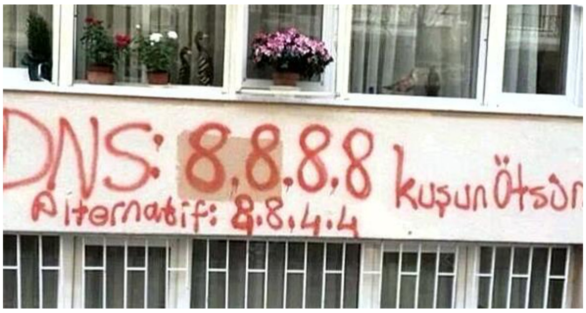

*Figure 1 – Graffiti in Turkey in 2014 in response to a DNS-based block on accessing Twitter*

The concept of so-called "clean feeds" has gathered some level of momentum in the DNS environment. These are DNS resolvers which make a feature of refusing to resolve DNS names associated with malware, abusive or criminal content, spyware and similar forms of abuse and threats. In this case the DNS is being used deliberately as a content filter by clients who make use of this service. Examples of such threat-informed DNS filtering include the Quad9 service (https://www.quad9.net/) and Cloudflare's malware-blocking resolver service at 1.1.1.2 (https://blog.cloudflare.com/introducing-1-1-1-1-for-families/).

There have also been various efforts to monetise the DNS or block the commercial activities of others by altering DNS responses. One of the persistent schemes to do this lies in NXDOMAIN substitution. Here the response to a query that would normally be NXDOMAIN, or "that name does not exist" becomes a response that references a search engine of some form. This gained prominence with the so-called "Sitefinder episode" in 2003 when Verisign, the registry operator for the .com and .net TLDs added a wildcard entry to these zones. This wildcard entry redirected users to a VeriSign web portal containing information about VeriSign products and links to partner sites. This wildcard entry meant that VeriSign effectively "owned" all possible .com and .net domains that had not been purchased by others and could use asll these orphan names as an advertising platform. It could be argued that in adding this wildcard entry to these zones Verisign was not in fact breaking the consistency of the DNS but was leveraging a DNS behaviour for their own ends. However, the technique has been copied many times since then and such imitations cross over the line of DNS integrity by substituting different responses instead of the NXDOMAIN response (https://www.potaroo.net/ispcol/2009-12/nxdomain.html). This was taken up by some browser vendors, who put their own spin on NXDOMAIN responses.

I suppose that at the heart of this question of DNS uniformity and consistency is the entire issue about infrastructure integrity. Aside from all the risks of viruses and worms and similar exploits of vulnerable applications on end systems, there are two major points of vulnerability in today's Internet that are insidious: routing and naming. Even when the end points are functioning perfectly well and are not compromised in any way, if the integrity of routing is compromised, or if the integrity of the DNS is compromised, then users can be lead astray quite unwittingly.

Lies in the DNS can be used to turn the Internet into a hostile place and turn users into unwitting victims. It seems to be bordering on hypocrisy for an ISP to claim that network infrastructure security is important and valuable, while at the same time deploy a name server system that deliberately generates lies in the DNS. Or are we supposed to be able to be able draw a distinction between "good lies" in the DNS, "harmless lies" and "evil lies"? If the ISP really wants to be treated as a common carrier, and not be held responsible for the actions of users and the content that users may exchange, then it should treat all user traffic with respect, and should refrain from acts of selective falsification of the DNS.

However, despite these arguments about defending the integrity of the DNS we have grown accustomed to such selective DNS blocks, and we have also grown accustomed to adopting measures to circumvent them as required. The justification of threat mitigation is one that is very persuasive in today's malware-ridden Internet and blocking the DNS resolution of malware command and control DNS channels is seen as a positive step in protecting Internet users. The issue of national regulation regarding content blocking references all the way back to the 1648 Peace of Westphalia and the sovereign right of nation states to impose rules on their subjects (otherwise crudely referred to in the twentieth century as a nation state's "monopoly on violence"). While there appear to be some common concepts of limits of unacceptable misrepresentation in the DNS (Sitefinder is still regarded as a step too far, for example), these various efforts on selective DNS lies and blocking appear to be commonly accepted as inevitable, just as the selective ability to circumvent them is also seen as a part of this space. It's something we have become accustomed to tolerating in the DNS.

Do we think that this level of deliberate inconsistency in the DNS is a current problem?

Evidently, we are not troubled by this observation, or at least not troubled enough to do anything about it in most cases. Such manipulation and selective filtering of the DNS is represented as a useful contribution to the unending (and at times unequal) struggle to cope with the incessant flows of malware, and while state-ordered DNS blocks may annoy some users, the means to circumvent such DNS blocks are sufficiently prevalent that is represents little in the way of an onerous imposition for users who feel affected by such moves.

So, while it happens, it's not seen to be a problem.

## An Open DNS?

Let's go back to the original question, and this time let's look at the DNS through the lens of the term *open systems*, as used in the computing industry.

What's an *open system* anyway?

It was a term that become popular in the 1980's as a reaction against vendor-proprietary systems. It is a term that was deliberately contrasted against the intentionally pejorative term of *closed* systems of that that time, where the systems developed by the vendor were constructed using private specifications that were not shared with third party vendors. Any and all third-party efforts to develop systems and services that interoperated with these proprietary systems or even went so far as to be plug-in replacements were actively discouraged by the vendor using all possible means, including resorting to litigation in order to protect their exclusive interest in the system's technology.

The original use of the *open system* term was associated with Unix, an operating system platform developed within AT&T at Bell Labs. In some ways the term *open* was somewhat inappropriate in this context, as while Unix was independent of any single vendor or host hardware, it was still proprietary code that was licensed to many vendors, including Microsoft, Sun Microsystems, Hewlett Packard and IBM. However, with continued use the term *open* became associated with a number of attributes of systems, including openly available specifications, the ability to implement systems or services against these specifications, preferably, although not necessarily, unconstrained by any IPR (intellectual property rights) claims.

Over time the term *open* was also associated with openly available source code, and available without charge. In other words, *open* as in *free*.

We can pose some questions about the *openness* of the DNS.

Is the DNS an *open system*? The DNS resolution protocol is an open standard published by the IETF without any constraints on access as to who can access this specification. By and large most of the DNS resolution protocol is free of any IPR encumbrance, although that is not a comprehensive claim relating to the entirety of the DNS specification. The process to change this specification is an open IETF process, open to all to participate in. I'd say that the DNS meets the criteria of being an *open* specification within the parameters we commonly use to define *openness*.

Is the DNS an *open-source* system?  There are a number of openly available implementations of DNS software, both as the server and the client forms of the protocol, so yes, it's an open-source technology.

Is the DNS an *open service* in term terms of anybody being able to access the name resolution service provided by the DNS system? If we deliberately constrain this question to the public DNS, as distinct from deliberately closed realms that we might find in enterprise or other deliberately closed environments, then once more we can conclude that the DNS is an *open service*. DNS authoritative name servers in the public DNS operate as promiscuous responders. They will proffer responses drawn from zones for which they are authoritative for to any client who presents them with a query. Anyone can access the public DNS. It's an *open* information space.

What about the content of the DNS itself? Is the information that is loaded into the DNS *openly available*? There is a subtle distinction to be made here about the entirety of the data that is brought together in a zone and the individual records in a zone. Individual records are freely available in the public DNS, available to any querier. The entire contents of a zone as a single corpus of data may not be open available, but I don't think that this impairs a conclusion that the information in the public DNS is *open*. The content of the DNS is not a secret.

Finally, DNS queries and responses are (or were in any case) open in the sense that they were not encrypted, nor were the IP protocol-related identities of client or server withheld from each other. Anyone with access to the path between a client and server could look at the protocol exchange on the wire and discern the query being made and the response that was provided. The DNS as a protocol was deliberately designed as an *open protocol*.

## When "Openness" is a Weakness

This final characteristic of the DNS, namely the use of an unencrypted protocol exchange between client and server to resolve a DNS name has become a massive issue for the DNS. Unlike the telephone network we do not use the Internet in terms of directly using end host protocol-level addresses (telephone numbers). We use the convenient level of indirection of making use of labels built upon words drawn from human language and using the DNS to translate these labels into protocol addresses. The implication is that pretty much every single Internet transaction starts with a call to the DNS.

What this implies if that if one could assemble the DNS transactions from an end user, complete with queries and responses then it would be possible to make a reasonably accurate profile of the Internet transactions made by that client, without being privy to those transactions. And, equally relevant in this network that is fuelled by surveillance capitalism, a DNS observer could assemble a protocol of each users' interest and activities and track the user in real time.

This is by no means a novel observation, and the power of the DNS to provide a comprehensive meta-data commentary on each user's activity has been known for many years (Figure 2).

THE RUMORS ARE TRUE. GOOGLE WILL BE SHUTTING DOWN PLUS—ALONG WITH HANGOUTS, PHOTOS, VOICE, DOCS, DRIVE, MAPS, GMAIL, CHROME, ANDROID, AND SEARCH—TO FOCUS ON OUR CORE PROJECT: THE 8.8.8.8 DNS SERVER.
Google

*Figure 2 – "Google Announcement" from XKCD (https://xkcd.com/1361/)*

Perhaps this level of openness is a weakness rather than a strength. A weakness not necessarily for the DNS, but for the rest of us, the Internet's users. We have little choice to use applications that make extensive use of the DNS, and the DNS is an open protocol that over-exposes user information and activities when is prone to third-party harvesting and exploitation.

Is the true cost of the Internet a comprehensive surveillance system that places each and every user into a fishbowl where every action is observed and remembered forever? Is this general corrosion of conventional expectations of privacy and trust necessarily a good thing for either the internet or our human society at large? More importantly perhaps is the question of how can we improve this situation? How can we draw a line on the openness of the DNS to provide some consistent level of respect for individual privacy?

## What is "the DNS"?

Let's start with the hardest question first: exactly what do we mean when we talk of "the DNS"?

What is the DNS as seen by an end user's device? From this perspective the DNS is a transactional network service. User applications send queries to the local agent (the local DNS resolver library on the end host), and this local agent returns a response (Figure 3).

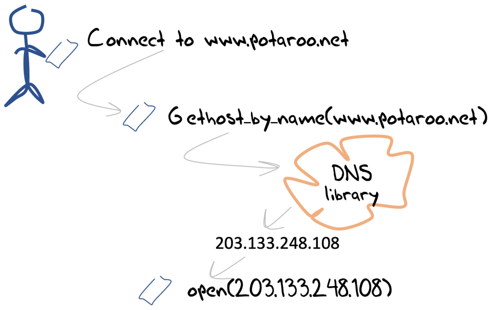

*Figure 3 – Host view of the DNS*

If you look a little deeper, you might observe that the local agent is passing its queries to a local recursive resolver. If you look at the incoming queries at an authoritative name server you might observe queries from recursive resolvers. This leads to a simple model of the internal arrangement of the DNS infrastructure where end users pass their queries to a recursive resolver. When the recursive resolver receives a query, the resolver first must work out whom to ask (discover the authoritative server for this domain name) and then direct a query to this server. The resolver will use this response to answer the original query from the client. The resolver will also cache the answer to allow it to reuse this information if it is asked the same query in the future (Figure 4).

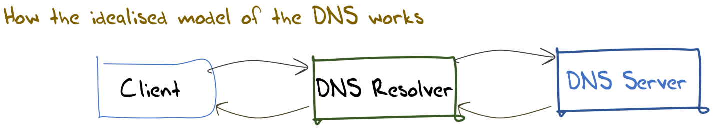

*Figure 4 –Idealised DNS Model*

As simple as this model of DNS name resolution infrastructure might be, it is also woefully out of date. The major factor in the Internet's three-decade public history is inexorable growth, and the DNS has certainly had to respond to these growth pressures. Single systems hosting recursive resolvers or authoritative services are no longer viable and these days we use a variety of mechanisms to defray the DNS load across multiple servers. Large scale services, both recursive resolvers and authoritative servers, are often implemented as a "server farm" where incoming queries are distributed across multiple server engines. There is extensive use of anycast to replicate these server farms in multiple locations, using the Internet's routing system to both distribute the load and in an approximate manner send the query to the closest service. What we see of the deployed DNS resolution infrastructure looks closer to Figure 5.

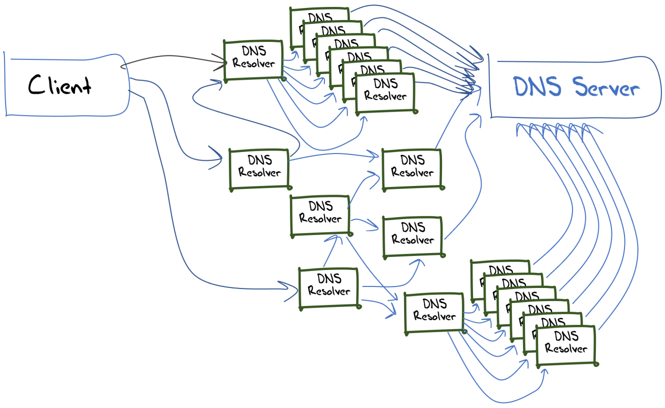

*Figure 5 –DNS Resolver infrastructure*

What this implies is that the DNS, when seen as a service, is not a simple set of centrally coordinated service delivery points. At best it is a highly distributed and very loosely coupled set of individually managed and operated components that leverages the fabric of the Internet itself to operate efficiently.

Perhaps this view of the DNS as a collection of interconnected service platforms is still an incomplete view. Maybe we need to step back and take a broader perspective. We use the term "the DNS" to describe various concepts:

- A **name space**. This is the collective name of a collection of word strings that are organised into a hierarchy of labels.
- A **distributed name registration framework** that assigns an exclusive "license to use" to entities for each unique DNS name.
- A **distributed database** that maps DNS names to name attributes.
- A **protocol** used by DNS protocol speakers to discover this mapping from a DNS name to an associated attribute value.
- A universally supported **signalling protocol**.

This multi-faceted view of the DNS leads to a conclusion that the DNS is not a single system, nor does it form a single market, open or otherwise. If we take this broader view of what the term "the DNS" encompasses, then there are many constituent markets which are, at best very loosely coupled. For example, these markets include:

- The market for registrars, who act as retailers of DNS names and deal with clients (registrants) and register the client's DNS names into the appropriate registry.
- The market for registries for new "top level" labels (gTLDs) operated by ICANN. This market is open to ICANN- qualified registry operators. A registry has an exclusive license to operate a TLD.

- The market for DNS name certification, which is a third party that attests that an entity has control of a domain name.
- The market for hosting authoritative name services.
- The market for DNS name resolution where users direct their queries to a resolver.
- The market for DNS resolver software
- The market for DNS authoritative server software

It's clear that the DNS is not a single marketplace, nor even a set of inter-dependent and tightly coupled markets with similar characteristics. Instead, the DNS has many component activities, some of which are highly regulated markets, such as the market for new top-level labels, others appear to be openly competitive, such as the market to host authoritative name servers, and some of which appear to be long term market failures, such as the vexed topic of DNS security.

## DNS Topics

Perhaps it's useful to use the DNS as a lens to observe a larger set of Internet topics, such as management of privacy, trust, cohesion and fragmentation, fragmentation and centralisation, abuse and disruption, and scaling. The reason why the DNS is a useful lens for all these topics is that the DNS these days is central to the concept of the Internet itself. We have successfully fractured the Internet's IP address space following the exhaustion of the IPv4 address pool and these days perhaps the best working model we have of what defines the Internet is the use of a common name space, namely the DNS.

That is way too large a scope of topics to considered here, so instead I'd like to develop just three topics in the DNS and look at them in further detail. These topics are trust, name resolution and fragmentation. They are important in my opinion because all three of these topics are potentially corrosive to the model of the DNS as the glue that is holding the Internet together, and these three topics show signs of incipient market failure, and its useful to understand how the DNS is managing to resist, or how it is succumbing to, these pressures.

### DNS and Trust

The DNS resolution protocol could best be described as a credulous protocol, designed in a different world where mutual trust prevailed. If a resolver sends a DNS query to the IP address of a server, then it will believe the response it receives. The problem is that this trust is completely misplaced, and the DNS system can be attacked to deliver faked responses where the client has no clear grounds to doubt the veracity of the response and no tools to test this assumption of veracity, so the client naively trusts every DNS data item it receives in response to queries it emits.

> Yes, TLS at the application level saves the day to some extent, and pervasive use of the TLS can prevent many forms of attempted misdirection, but it's useful to appreciate that TLS does not provide an assurance that you are indeed reaching the service you intended to reach. TLS only provides a more limited assurance that you are reaching a site via a DNS name where at some time in the past someone was able to demonstrate to a Certification Authority that it had some form of operational control of that same name. If an attacker was able to deceive a Certification Authority, then the subsequent assurances provided by TLS for the attacked DNS name are generally void.
>
> Saying "I trust that this service is genuine because TLS says its good" is in fact saying "I trust that this service is genuine because someone whom I have never met, has, at some indeterminate time in the past, conducted some tests, of which I have not been privy to, and been satisfied enough to associate a private key with this DNS name." By any reckoning TLS is presenting a bizarre definition of "trust" when put like that!

Obviously, this trust can be abused in many ways, and much effort has been spent in devising ways to test the veracity of a DNS response. We'd like to be assured that a DNS response is an accurate copy of the original zone data, and the response is current (as distinct from a replay of stale information).

In response to the threats posed by this level of credulity in the DNS resolution process we've devised a framework of digital signatures that can be attached to DNS responses that attest to the accuracy, completeness, and currency of the DNS response, called DNSSEC. The method of attaching this digital signature does not alter the behaviour of the DNS protocol. Nor does it change the behaviour of DNS servers or caching DNS resolvers. When a sone is signed with DNSSEC each resource record in the zone has an associated digital signature in the form of additional Resource Records, and a canonically sorted list of records is linked with NSEC (or their hashes with NSEC3 RECORDS). DNSSEC-validating clients request these additional Resource Records and perform additional checks to verify the digital signature associated with a DNS response (Figure 6). Clients discard the DNS response if the validation process fails.

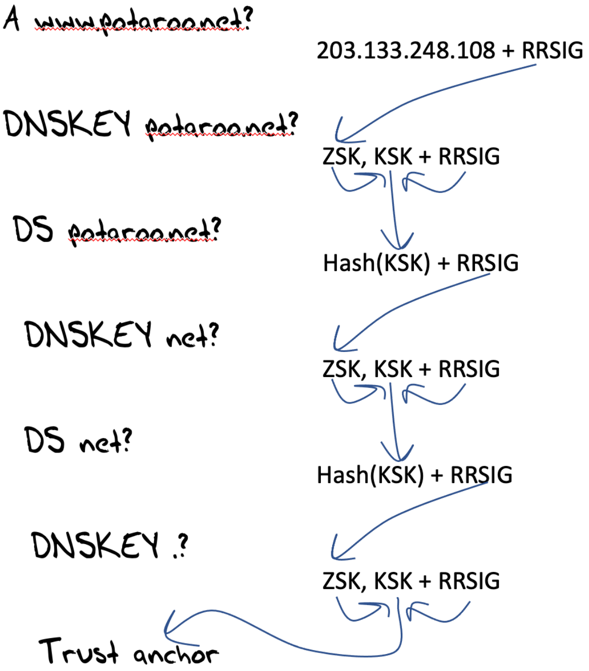

*Figure 6 – DNSSEC Validation process*

You would think that an incremental change to the DNS that improved the level of trust in DNS responses would prove to be immediately popular. It would be far harder to use the DNS to deceive and mislead end users and you might think that end users would be all in favour of a move that increased the resilience of their online environment. And if it's popular with end users then you would also think that it would be popular with DNS providers. And if DNS resolver clients are performing DNSSEC validation then you might think that zone authorities would be sufficiently motivated to DNSSEC-sign their zones, as this would allow clients who query for these names to detect if their DNS resolution of these names is being tampered with.

You might think all that, but you'd be wrong if you did. Progress with the deployment of DNSSEC signing and validation has been extremely slow by the admittedly impatient benchmarks of the Internet (Figure 7). DNS zone admins appear to be generally reluctant to DNSSEC-sign DNS zones, and DNS resolvers appear to be similarly resultant to DNSSEC-validate the DNS responses that they process.

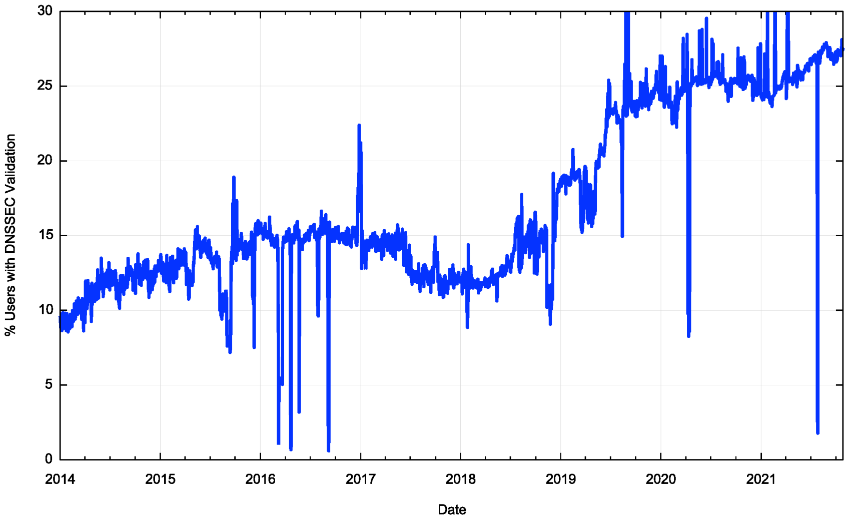

*Figure 7 – Uptake of DNSSEC validation over time (from https://stats.labs.apnic.net/dnssec)*

Why is the adoption of DNSSEC taking so long? What is the threshold of critical mass where further uptake would be motivated by the size of the already-adopted user base? If we have yet to reach this threshold then it's possible that failure of DNSSEC is still a very real option. Why do we find ourselves in this indeterminate position?

It's certainly true that DNS resolvers must work harder and take more time to perform DNSSEC validation. They must perform a full back trace query sequence all the way through the delegation hierarchy to assemble a fully interlocking DNSSEC signature chain from the root to the name being validated (Figure 6). This sounds far worse than it normally is, because, as with many operational issues with the DNS, caching papers over many potential performance issues and makes the validated resolution of popular names exceptionally fast. However, the DNS responses may also get larger because of attached signatures, and large DNS responses cause further performance issues in the DNS. These large DNS payload issues cannot be simply solved by caching. There is a time and efficiency cost for DNSSEC-signed zones, and the Internet is both exceptionally intolerant of DNS delay and highly reliant on DNS efficiency.

In terms of protection of the end user DNSSEC is only half implemented. The final resolver, the DNS stub resolver on the end host normally does not perform DNSSEC validation itself, and simply relies on the recursive resolver to perform the validation on the user's behalf and set a bit in the response given to the stub resolver to indicate that the response has been validated. Given that this stub-to-recursive path is typically an open unprotected path, the assurances provided by this simple one-bit signalling mechanism are exceptionally weak.

DNSSEC also imposes costs and constraints on the zone publisher. The zone publisher can elect to generate a complete ordered zone, generate a signed version of this zone, and pass this signed zone to the DNS authoritative servers. However, this pre-signed model supposes that the entire zone can be assembled in one location at a point in time, which may be operationally challenging for very large zones with highly dynamic content. With the increasing use of customised behaviours in the DNS (where the DNS is used to provide a "geographically nearest" answer to a query, for example) this statically signed DNSSEC model is overly constraining, but the alternative, namely dynamic signing models that attach signatures at the time the response is formulated, impose an incremental load on the zone's authoritative servers, and the issue of dynamic processing of signed no such name (NXDOMAIN with NSEC(3)) has to be addressed.

Bootstrapping a DNSSEC delegation is challenging. Once the delegation point (parent) and the delegated zone (child) are securely linked via the DS record, then further operations, including keyroll can be performed securely, but the initial installation of the DS record in the parent requires some level of trust on first use, or the invention of an entirely new framework of secure credentials to pass this information from child to parent. Neither approach is all that satisfying.

If the costs of creating a trustable DNS system where the information in the DNS can be passed across the Internet while maintaining trust in the integrity of the DNS data are so high, then will we ever get to the point that the DNS is trustable? And if we cannot rely on the DNS to provide a complete and accurate view of the underlying name space of the Internet, then is the resultant compromised DNS something we should rely on to the extent that we do today? Are we relying on the integrity of the Internet's name system because we have no alternative?

It can be argued that these failings of DNSSEC leads to the observation that DNSSEC is a market failure which in turn poses some very uncomfortable questions about the DNS itself and the larger issue of trust in the Internet environment.

End users, the ultimate beneficiary of the DNS system, don't pay to pass queries into the DNS resolution system, nor do they pay for responses. Their expectation is that the DNS is a bundled component of the ISP service that they are using. With specific reference to DNSSEC, users have no leverage with recursive resolvers in terms of expressing a preference for authenticity in DNS responses. They have no choice in what names they query for, in so far as they have no ability to express a preference for using domain names that are DNSSEC-signed.

Recursive resolvers are typically provisioned as part of the bundled service provided by an ISP, and users to not normally make ISP choice decisions based on the relative quality of the bundled DNS service. There is little in the way of financial incentive for an ISP to invest in the resolver, particularly in terms of including DNSSEC validation. It's a cost without visible user benefit that is expressible in the form of user choice of ISPs.

Authoritative servers should be highly motivated to DNSSEC-sign their served zone, but without widespread adoption of DNSSEC validation by recursive resolvers and end user stub resolvers then it's challenging to justify the incremental costs associated with serving a DNSSEC-signed zone.

It's apparent that the costs of DNSSEC are associated with the publication of a signed zone and the recursive resolvers' processing of signed information, while the benefits of DNSSEC, such as they may be, accrue with the end user. Because the end user is largely unable to express a preference for DNSSEC in terms of the DNS names they use or the recursive resolvers that handle their queries, then the incremental benefit realised by the end user in being able to place trust in the integrity of these responses to queries of the Internet's name space is unable to be directly or indirectly expressed to the parties incurring the incremental cost of DNSSEC. Cost and benefit are mis-aligned in this space, and this lack of alignment leads to confused signals and a very fragmentary adoption process, as we see with DNSSEC adoption so far.

Despite its importance and the lack of any viable alternative technology DNSSEC appears to be a market failure, and in the loosely coupled environment of the DNS where market signals become the primary means of orchestration between providers and consumers within the various activities in the DNS, failures of this form of market signalling have serious consequences. If we are collectively incapable of taking the necessary steps to instil required levels of trust in the DNS, then the obvious response from those whose need is most pressing is to adopt a more piecemeal approach and carve out parts of the DNS where it is possible to develop high trust spaces within those deliberately segmented parts. If this sounds a lot like DNS fragmentation, then that's not a coincidental observation.

## DNS Resolution

In the DNS environment Recursive Resolvers sit in the "middle ground" between clients who want to query the DNS and authoritative servers who can answer queries about the information contained in their zone (Figure 4). Recursive Resolvers perform a discovery phase by undertaking a top-down iterative search through the hierarchically organised DNS server space progressively finding the authoritative servers for each successive lower level in the server hierarchy until they establish the authoritative servers for the zone who can answer the name query. All this would be tortuously slow, but the recursive resolver caches what it learns from DNS responses, and over time much of the overhead of this iterative name server discovery process is eliminated by using the local cache.

Being in the middle ground in the DNS also implies that resolvers lie in the middle ground of funding. Clients don't pay resolvers a per query transaction fee to handle their query so there is no revenue stream from the client side. Equally, servers don't pay resolvers a per query transaction fee to process and cache their response so there is no revenue stream from the server side.

The way this has been addressed within the Internet service model is that the access ISP (or retail ISP) bundles the service of DNS recursive resolution service into the access fee charged to clients. Because the quality of the DNS service is not normally a competitive distinguishing feature for clients, the DNS service is typically regarded as a cost element in an ISP's activity profile, and it receives minimal investment and attention as a result. This, in turn, implies a minimal budget for DNS resolver software so the DNS resolver market is heavily biased towards various freeware solutions.

More critically, there is strong resistance from these resolver operators to support changes to the operation of the DNS that would incur significantly higher cost unless there was a comparable additional revenue stream. It has been estimated that a shift in the DNS resolution protocol from UDP to TCP would reduce the capacity of a resolver to manage a query load to some 30% of the UDP-based capacity. If the entirety of the DNS traffic made this shift to TCP, then to maintain the current service levels the platform capacity requirements would triple, with a comparable call on capital funds to provide this increased capacity. It is not entirely clear at this stage what incremental costs are incurred by the platform when the DNS payload is encrypted, but there is a general impression that the greatest cost increment lies in the shift to TCP, and the additional encryption costs are relatively lower. The bottom line for many ISPs is that DNS resolution is a cost element, not a revenue source, so any change in the DNS that would increase their costs in providing this DNS resolution service can be expected to be passively resisted.

There have been various efforts to change this picture through efforts to attempt to monetise the DNS resolution service, but these efforts have been, in general, unsuccessful. Various forms of funded misdirection through "sponsored" DNS results that are substitute responses are view with extreme disfavour, and slightly less obnoxious efforts that substitute search engine references instead of "no such domain" NXDOMAIN DNS responses also attracts much censure. Altering DNS responses outside of government direction or malware filtering is not well regarded in the DNS. The other monetary temptation is to take the resolver's query log and sell it. As tempting as this may be as a revenue source, it is often not permitted within a national communications regime. Increasing levels of user sensitivity over such covert intrusions into users' expectations of privacy also appear to act as a deterrent to such activity, even it if is permissible within the local regulatory framework. The result is that an ISP's DNS resolver service often is the victim of operational inattention, out-of-date software and underperforming service.

These ISP-operated DNS resolvers have also been perceived as a source of problems in the overall Internet user experience. Whether this is actually the case or not is largely speculation, as individual ISP-provided DNS services vary greatly in quality, but the perception is a persistent one in any case. Application design and protocol engineering both obsess on shaving milliseconds of extraneous delay in Internet responsiveness. All of this considerable effort to tune the application experience is completely negated if a DNS query takes the ISP's DNS resolver multiple seconds to process and respond.

One response to this situation is for third parties to provide a DNS resolver service and allow users the choice to bypass the ISP service and use this third-party service. It may be the case that the third-party service operates in a different national jurisdiction and is not subject to the same national constraints regarding serving certain DNS names or provides a different geolocation signal to a service provider which in turn could circumvent geolocation controls applied to content, in those cases where the DNS resolution process is used as the primary geolocation indicator. It may also be the case that this third-party service is more reliable or simply faster.

Other motives relate to the third-party open resolver provider. NXDOMAIN substitution leads to a user behaviour of using the DNS as a search engine selector, bypassing the default choices made by the browser or the platform. Search engines act as an indirect generator for significant revenues (just ask Google!) and search providers have been known to fund browser platforms and others to make their search engine the default in such platforms. NXDOMAIN substitution undermines the platform's default search engine selection and exposes the user to a search engine nominated by the DNS-provider. By offering a resolution service that does not perform such substitution, then the platform defaults for search substitution are restored.

There is also the prospect of the third-party resolver operator performing data analysis on the DNS query stream and monetising the results. Because the open resolver operates at the application level rather than as a network access provider their position with respect to telecommunications provisions and privacy requirements is arguably less well defined in many regulatory regimes. While this potentially unconstrained access to data may have been an interesting position some years ago, two factors appear to have closed this down. The first is legislated moves to force all service providers to exercise far greater levels of care in the handling of all aspects of personal data, such as the GDPR provisions in the EU. Even if the client's IP address is not part of the captured DNS query, the captured query data is a very rich vein of personal data and it's likely that this query data is protected under such legislated provisions. The second factor is an increasing user awareness of such channels of data acquisition, and a visible aversion to the practice. All such services these days require more explicit practice statements that both describe the providers practices and bind the provider to adhere to them, and often require the user's explicit permission to gather such data. In situations where users are given a set of equivalent services only some of which also include provisions for user profiling and on-selling such profiles it has been a common experience that users tend to choose the alternatives that explicitly eschew such data collection practices.

The common issue with these third-party open DNS resolvers is that they are essentially unfunded by the clients of the service. This is an ideal condition for a success-disaster, as the greater the level of use of these services then the greater the capability of infrastructure required to service the query load, which calls for ever greater levels of funding.

Redirecting the way DNS is managed has historically been the preserve of a small collection of technically adroit users who are sufficient motivated to customise their DNS resolution environment. This is not a high volume of queries, and the third-party DNS resolvers were also provided by this same community on a volunteer basis. It was a small-scale operation that had little impact on the mainstream of the DNS. But all that has changed in recent years.

Looking at the Internet and its user base today, the default position, used by some 65% of the Internet's user population, is to pass their DNS queries to the ISP's recursive resolver service, and the remaining 35% of users have their queries passed to a recursive resolver that is not part of the ISP's local infrastructure (Figure 8).

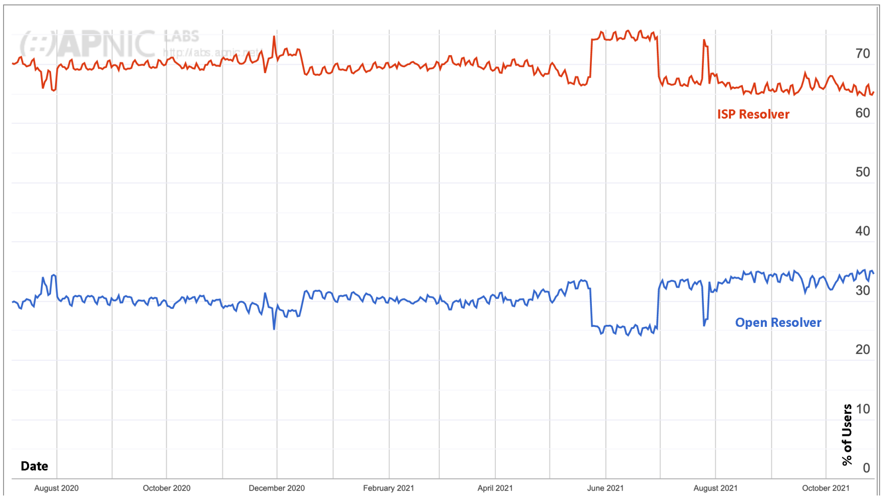

*Figure 8 – Use of Open Resolvers in the DNS – from https://stats.labs.apnic.net/rvrs*

This 35% is perhaps a misleadingly high number. Individual end users, on the whole, do not deviate from the default DNS configurations provided in their devices and provided on their local network. In some cases, the device has embedded the DNS configuration so deeply that it cannot be changed in any case, and in other cases the user is unwilling to make such changes dues to unknown implications if they were to do so. So in many cases it's the ISP itself that is redirecting queries to open resolvers.

Why would an ISP do this?

Probably because it's cheaper, and as we've already considered, from the perspective of the ISP the DNS resolver service is often seen as a cost element without contributing to competitive service discrimination in the eyes of the user.

We see in a number of economies, and Chad in Africa is a good example here, where entirety of the country's user population is observed to make their DNS queries via Google's public DNS service (https://stats.labs.apnic.net/rvrs/TD). This would be the case if the ISP's own DNS service is constructed using a simple DNS forwarding construct. The ISP does not need to invest in the expertise and equipment to operate a DNS service, and presumably Google provides a functional and efficient DNS service that is free of charge to the ISP.

It's not just ISPs that perform this DNS forwarding function as a way of defraying costs. Enterprises also appear to make extensive use of these third-party open DNS recursive resolvers, presumably due to issues relating the resilience and consistency of the DNS service.

This discussion of the third-party DNS resolution space might lead you to conclude that this is a conventional market with a number of providers who are competing for clients across the Internet. This is not the case. Google's 8.8.8.8 Public DNS service dominates this space, and close to 30% of users have their DNS queries processed through Google's service. That's the single largest DNS resolution service across the entire Internet. In the third-party DNS resolution space Cloudflare's 1.1.1.1 service is used by 4% of users, and the OpenDNS service, now operated by Cisco, is used by a little over 2% of users (Figure 9).

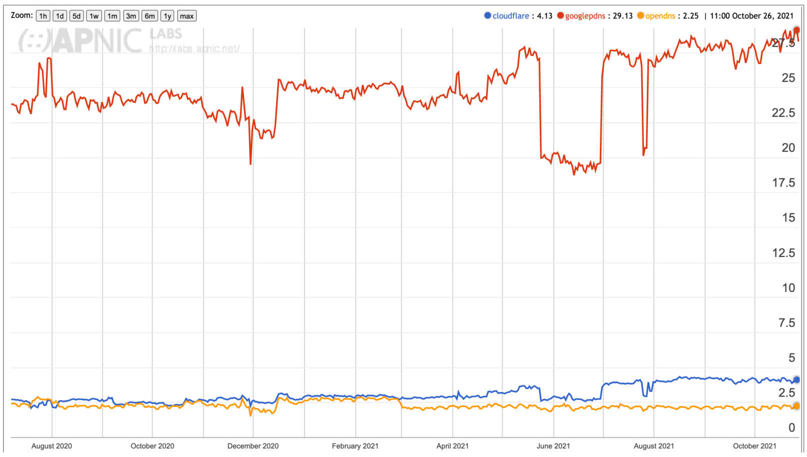

*Figure 9 – User share of the 3 largest open DNS resolvers DNS – from https://stats.labs.apnic.net/rvrs*

This highly dominant position by Google in the DNS raises many serious questions about the DNS. The recursive resolver is the binding between client and provider, and the observation that close to one third of the Internet's user population is passing their queries through Google's DNS service gives Google a pre-eminent position within the Internet. If this service were to deny the existence of a DNS name, or an entire name hierarchy, then for up to one third of the Internet's users the name would no longer exist. Google's service performs DNSSEC validation, but end clients do not. If this service were to misrepresent a resolution outcome, then the same end user population would believe the lie. Even if the name was DNSSEC-signed the lie would be undetectable by the end users. By handling the DNS query traffic for up to one third of the users on the Internet, and by having an unparalleled view of the end-user clients of this service Google also has perhaps the richest observatory of activity on the Internet, at a level of granularity that extends all the way down to individual users.

I should hasten to note that these are questions about the potential for abuse of Google's unique position in the DNS and no more. Google are always careful to clearly state that their service does not lie or misrepresent information in the DNS. If it's in the DNS then a query to Google's service will show it. If it's not in the DNS then Google's service will return NXDOMAIN (https://developers.google.com/speed/public-dns/faq). Google are also clear about their privacy policy with respect to DNS data. The data that is gathered from this service is only stored for a limited period and not shared. Sampled data at an aggregated level is stored for longer periods (https://developers.google.com/speed/public-dns/privacy). Google will expose information in response to certain legal and governmental requests, and report on these requests in a transparency report (https://transparencyreport.google.com/user-data/overview).

On the other hand, Google's service has had an interesting positive impact on the ability of the DNS to overcome its seemingly structural resistance to change. DNSSEC is having its problems in gaining acceptance in the DNS, but without Google's early commitment to support DNSSEC validation it's likely that DNSSEC would simply not exist. It is perhaps no coincidence that there is a high correlation between the count of users who use resolvers that perform DNSSEC validation and the count of users who have their queries sent to Google's DNS resolver service. The map of deployment of DNSSEC validation in the Internet is largely a map of the use of Google's service. It is possible that other technologies, such as support for DNS queries over TLS and HTTPS channels is going to gain acceptance solely because of Google's willingness to support such query models in their Public DNS platform. Query Name Minimisation, aggressive NSEC caching and other techniques that minimise gratuitous information leaks from the DNS are all part of Google's service. Similarly, the avoidance of UDP fragmentation in large DNS responses are part of Google's platform. The use of prepended nonce labels to increase the entropy in queries as a means of mitigating off-path guessing attacks is a Google-adopted measure (https://developers.google.com/speed/public-dns/docs/security).

The current situation with Google's central position in DNS resolution is a position that has an undeniable element of critical dependence on a single provider, and we are dependent on the undertaking of that provider to operate in the common public interest and not to exploit their unique position for their own ends.

This is not without historical precedent and the undertakings made by Theodore Vail to the US Congress in the Kingsbury Commitment of 1913 struck a similar tone of self-restraint on the part of the emergent AT&T. AT&T undertook to moderate its behaviour to allow for continued competition and upholding a broader public interest in the private sector operation of a public utility. This undertaking to sustain a competitive market for local phone services did not even last for a decade, and in 1921 the Willis Graham Act condoned AT&T's defacto monopoly position. The entire concept of the enlightened private sector utility operator upholding the public interest was quietly abandoned at the time.

It would be good to think that this time it will be different, but we have no real evidence to back up this hope.

In terms of the longer-term health of the DNS as a central piece of infrastructure for the Internet, its independence and neutrality in the competitive landscape is essential. Yet this is exactly what is of concern here due to this over-arching position of Google's in the name resolution function.

### DNS Fragmentation

The earlier discussion on DNSSEC concluded with the observation that the obvious response from those whose needs are most pressing is to adopt a more piecemeal approach and carve out parts of the DNS where it is possible to develop high trust spaces within those deliberately segmented parts. If this sounds a lot like DNS fragmentation, then that's not a coincidental observation.

Let's look at this aspect of fragmentation in more detail, as it's not just DNSSEC where the pressures to implement changes in the DNS are unequally felt.

This has also been visible in the consideration of the issues of changing the transport protocol in the DNS away from unencrypted UDP to encrypted sessions running over TCP or QUIC. There are a couple of pressures that are pushing the DNS away from UDP. The first is that the DNS's open protocol has been, and continues to be, widely abused. The DNS is a rich window on the actions of each user and assembling the set of DNS queries made by a user is functionally equivalent to observing the activities of the user. This observation does not require spyware on the device, nor does it require eavesdropping within the local network. If the eavesdropper can see the query stream as presented at the recursive resolver, then this is sufficient. It would be hopelessly naive to think that this never happens, and it is far more reasonable to believe that ISPs would duly surrender excerpts of their DNS logs to empowered authorities when presented with the duly executed warrants. Cynically, one could also believe that the presentation of a warrant is an optional extra in this process within many regimes at various times.

The second consideration is that the DNS is an amazingly effective attack weapon. An attacker can present a DNS resolver with a steady stream of small DNS requests over UDP and it will receive a steady stream of DNS responses. Because it's using UDP it's not possible to validate that the source address of these queries is genuine, and all an attacker need do is to use the address of the intended victim as the source address of these spurious queries and that's about it. Much effort has gone into trying to make DNS resolvers and servers more resistant to this form of attack, while a similar evolution has been undertaken by attackers to present a query profile that is not immediately recognised by the servers. The underlying problem here is the use of UDP in the DNS, and a more effective response is to stop using UDP and place DNS traffic over a two-way protocol exchange, such as TCP.

If we are looking at an encrypted DNS exchange over a two-way protocol, then surely DNS over TLS is a good-enough answer? Yes, that's true, but if we are going to perform this transition then why not got just a little further and push the DNS into HTTPS exchanges. What will we gain with this additional step? Firstly, we leverage the Internet's universal support for HTTPs, and secondly, and perhaps more importantly, we can push the DNS away from common platform libraries on the edge device and make it an attribute of the application. Why would we want to do this? Because applications have a far better idea of what they are about to do than platforms. Applications can pre-provision DNS responses in advance of user direction, loading all possible outcomes of a click before the click is made.

However, it's more than just being faster at anticipating the user and making the Internet appear to be ever faster. We can leverage the observation that an application running on a device is in fact the outcome of a shared state between the client device and the application server. What if the application server becomes the DNS resolver for the application running on the client device? Can the server send DNS results directly to the client application before the client even sends the DNS request? Obviously, of course it can. Why would it do so? Because the resulting user experience is even faster.

At this point in this speculative exercise we've left a lot of the existing DNS infrastructure behind. We've dispensed with common DNS stub resolver libraries on platforms and replaced them by functional modules embedded in applications. We've dispensed with the model of common recursive resolvers that process all DNS queries and pushed the DNS queries and responses into the client/server interaction that happens within each application context. We dispensed with the query/response model and allowed application servers the ability to anticipate the actions of the application's clients and pushed information to the client before its needed. There are a couple more steps we could take here.

The first step is to increase the richness of the DNS responses. The IETF is in the process of completing the specifications for the SVCB and HTTPS DNS resource records. This is a big step away from the simple mapping of names to IP addresses into a tool that can answer the set of questions around the concept of

"How can I connect to the "foo" service?" These connection questions include: What transport protocol can I use? What port numbers? What encryption profile? What keys? What IP addresses? What are my alternatives? The desire here is to retrieve all of this service rendezvous information in a single DNS response, retrieved with just a single round trip time of delay. We've already seen with the much maligned (and rightly so) Client Subnet option in DNS queries that the DNS is being used to customise its answers based on information provided in the query. So, it's not a big leap for the DNS to not only carry this connection profile to complete a rendezvous with a named service, but to customise this profile for the querier to provide the "best" instance of a service to meet the specific requirements of the client.

The second step is to observe that at this point when the DNS has been shifted into being an application-level signalling channel there is no particular need to strictly adhere to the confines of the public DNS as a name space. Applications can add to this space with additional named items that only exist in the confines of a server/client interaction for this application. Because the channel between server and client is a constrained and private channel this is hardly a big step for the application.

We are now a long way from the DNS we're all familiar with.

Why would we want to go down this path?

We don't all want to do this. But for those who do, their answer is to fulfil a desire to increase the agility, flexibility and responsiveness of application behaviours leveraging DNS technology as a universally accessible distributed data directory without the application world being forced to coordinate the investment in extensive upgrades to the current DNS infrastructure base. The world of content and services is impatient for change, while much of the existing DNS infrastructure is stuck in a status quo because it's starved of resources because it does not directly generate revenue. The resolution of these conflicting pressures is most likely to be found in various forms of fragmentation of the DNS.

It appears likely that applications that want to tailor their DNS use to adopt a more private profile will head off to use DNS over HTTPS to access an application-selected DNS environment, leaving the device platform as a legacy service using unencrypted DNS over UDP to the ISP-provided recursive DNS resolver and the remainder of the DNS infrastructure. That way each application ecosystem can deploy their own name and signalling infrastructure and avoid waiting for everyone else to make the necessary infrastructure and service investments before they can adopt these mechanisms in the common DNS infrastructure system.

The imbalance in the financial markets where the market capitalisation of the so-called big tech companies completely overshadows all other enterprises is echoed in various forms of imbalance in the infrastructure of the Internet. Those parts of the Internet environment with sufficient motivation and resources will simply stop waiting for everyone else to move. They will just do what they feel they need to do!

The prospect of a fractured DNS is a very real prospect right now.

## An Open DNS?

A truly open technology is not a museum artefact, fixed and unchanging.  A truly open technology is as open to innovators as it is to adopters and users. At times the conversations between the various stakeholders may be uncomfortable, particularly when there are differing objectives. While committees may attempt to resolve such differences by creating odd compromises in an effort to address part of the desires from each stakeholder, markets tend to be more direct and uncompromising in their outcomes, typically resolving the tension in favour of one party over the other. The is the evolutionary process we are seeing in the DNS, which itself reflects broader pressures we are seeing in the Internet at large. This deregulated environment which uses market forces and user preference as their driving impetus is often uncompromising in its winner-take-all approach.

The overall progression here is an evolution from network-centric services that leverage transmission capabilities through a phase of platform-centric services (does anyone still remember Windows 95?) to today's world of application-centric that support a rich world of replicated content and services. It's clear that the DNS is being swept up in this shift, and the DNS is subject to strong pressures to adapt to this environment.

Will the DNS emerge from these tensions with a single unified coherent name space intact? Or will the various pressures result not only in a fragmented DNS platform, but a fragmented name space as well?

I have no answer to these questions, and nor does anyone else right now. We can only wait and see how it all plays out within the parameters of the DNS and the Internet at large. However, I suspect we won't have to wait very long as the impatience for change has been growing on the part of the upper layers of the protocol stack. The more capable the end devices become then the greater the level of function and autonomy we can stuff into the applications that they run. In fixating on the end user experience, we are now looking at how to create blindingly fast and rich environments, and in trying to achieve that we are losing collective patience with the ponderous state of evolution of common infrastructure in the Internet. Rather than wait for all those DNS libraries and stub resolvers embedded in CPEs to be upgraded or wait for ISPs to improve their recursive resolver services, today's response appears to be a determined effort to drive around the problem and take the relevant bits of the DNS along with these changes to application behaviour.

Undoubtedly the DNS is changing. But in all this I would suggest that these changes are a clear indication that the DNS is still "open" in the sense that it is open to further change and transformation. I would suggest that this is a good thing for both the DNS and the Internet, in whatever forms they may assume in the future.

## Disclaimer

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

## Author

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

*www.potaroo.net*