December 2021
Geoff Huston

ICANN DNS Resolver Symposium

ICANN hosted a Resolver Operator Forum in mid-December, and the session had several interesting presentations that I would like to comment on here.

DNS Resolver Evolution

The first presentation in this forum was from Paul Mockapetris. He pointed to the original academic published paper, *Development of the Domain Name System*, by Paul Mockapetris and Kevin Dunlap, published in the proceedings of ACM SIGCOMM'88. The paper noted that by 1983 it was obvious that the shared HOSTS.TXT file was not a scalable solution, particularly as shared mainframes were changing to personal computers at the time, and it described the initial efforts to change this name-to-address mapping function into a distributed database that it hoped would allow scaling of the system. The paper adopted the approach of identifying the notable successes, failures, and surprises in the process.

DNS in the late '80's

It is certainly interesting to look back more than thirty years and identify what was seen at the time as a readily mutable implementation setting become cast in concrete over the intervening years. For example, the paper notes that "Labels are limited to 63 octets and names are restricted to 256 octets total as an aid to implementation, but this limit could be easily changed if the need arose." This limitation is firmly in place today.

It has been observed that for some time that the proposition of queries seen at the root servers for non-existent domain names is persistently high - in excess of 70% in some cases for the root servers. The paper voiced an entirely different expectation: "We expected that the negative responses would decrease, and perhaps vanish, as hosts converted their names to domain-name format". This expectation is yet to be achieved.

On the issue of caching: "Our conclusion is that any naming system that relies on caching for performance may need caching for negative results as well. [...] This feature will probably become standard in the future." And this subsequently happened.

On the choice of UDP: "The use of datagrams as the preferred method for accessing name servers was successful and probably was essential, given the unexpectedly bad performance of the DARPA Internet." The use of UDP today is seen as a critical factor in scaling the DNS, allowing servers to handle large query loads without excessive consumption of system resources. The issue with UDP's lack of implicit acknowledgement did create some concerns with retransmission behaviours, and the paper noted that "Much unnecessary traffic is generated by resolvers that were developed to the point of working, but whose authors lost interest before tuning." The issue of UDP packet fragmentation was not seen as an issue at the time: "The restriction to approximately 512 bytes of data turns out not to be a problem." It should be noted that the predominate MTU size in the Internet in the mid 1980's was 576 octets.

In what was described as a success was the observation that caching works extremely well: "The caching discipline of the DNS works well, and given the unexpectedly bad performance of the Internet, was essential to the success of the system."

A failure in the original model, which Paul asserts continues through to today, is that data types are hard to create (although the recent experience with the SVCB and HTTPS data types appears to point to a different conclusion, that data types are relatively easy to create!). As Paul suggests perhaps it was the overly complex IETF registration procedures that were pushing DNS users to overload the TXT record as an alternative to creating more data types.

Some original "features" of the DNS protocol were never implemented. Queries have a 16-bit query count field, and there was provision to stack multiple queries into the same DNS protocol data unit, theoretically allowing for a set of queries in a single transaction. The query count value has always remained at 1.

Perhaps the biggest failure at the time that persists through to today was the decision to use a restricted ascii character set, with equivalence of upper- and lower-case alpha characters. In retrospect it appears that the protocol would've benefited from an exact binary match algorithm, leaving character equivalence as an application function rather than an intrinsic protocol attribute. Paul had thought that the label itself might, in some way, dictate the equivalence algorithm to use over DNS labels, with the system itself would transparently handle binary data. The result has been that we are attempting to retro-fit Unicode binary labels into this restricted character set with some surprising challenges.

Fast Forward to 2010

The major shift in the DNS architecture in the intervening two decades was the step of interceding a network between the client and the resolver. DNS recursive resolvers no longer reside on the same platform as the applications that call for DNS resolution, and they are connected not by any form of RPC (Remote Procedure Call), but by the DNS protocol itself. This has been both an amazing success and a critical vulnerability. The success was this this has allowed the DNS infrastructure to continue to scale to meet ever-increasing demands, and to do so without being constrained by other infrastructure investments. The DNS was not an intrinsic part of the underlying network, nor was it an intrinsic part of the server or client platforms. The intrinsic elements of failure in this model are based on the use of a simple open transaction protocol in the DNS, which was available for inspection by any well-positioned observer and could also be manipulated by active intermediaries.

The underlying vulnerability here was that the DNS query structure is both recursive and stateless. If a resolver cannot directly respond to a query from its local cache, it will make its own queries, these “triggered” queries are made without any reference to the original query and include no details of the original querier. When the resolver responds to the original querier there is no indication how the resolver gathered the information, whether by use of the local cache or by performing further queries. This makes the DNS opaque, in the sense that the original client has no ability to trace the queries being made on the client’s behalf and has no way to understand how the resolver has assembled the information to generate a response.

This leads to what Paul describes as a notable failure of the time, namely the exercise to retrofit verifiable authenticity to the DNS in the form of DNSSEC. It should be stressed that DNSSEC is not in itself a failure in terms of the ability of a user to validate a response, as long as the original DNS zone has been DNSSEC-signed and there is a clear linkage of signed zones from the target zone all the way back to the root zone. The failure has been in the manner of its integration into the DNS, in that it has been bolted onto the DNS without any changes to the base DNS protocol. More information needs to be passed through the DNS, which adds to the size of DNS responses and increases the number of queries required to complete validation. Both behaviors are an anathema to the efficiency of the DNS, and the result is that the timelines for the adoption of DNSSEC-signed zones and the adoption of validation, both at the recursive resolver level and at the end client stub resolver level, is at best sluggish. In si\ummary, the situation with DNSSEC is that depending on your point of view there is either too much of it, or not enough of it, and we can't agree as to which is the "right" answer!

What has been surprising is that the DNS has been used as a policy control point. To counter the spread of malware, spam and similar, it has been effective to simply block the resolution of the associated DNS names in the DNS. This started as simple "blacklist" for a resolver to consult before attempting resolution and has been subsequently standardized as a policy construct in Response Policy Zones (RPZ). These DNS controls are not only commonplace, but have been taken up my many actors, from governments to open DNS resolvers.

Another surprising outcome in developments in resolvers that some original simple principles of DNS resolution that we knew in the 1980's are too often ignored in recent implementations. A case in point is the so-called "Tsunami" vulnerability in the DNS where a circular chain of references in the DNS may cause a resolver to chase the loop for an extended time. This vulnerability has been known since the inception of the DNS and the simple principle from the 1980's to defend against such situations is to limit the total sum of resources that a resolver will use to work on any query.

Fast Forward to Today

The increasing distance between the client and their recursive resolvers and opened the DNS to all kinds of privacy incursions, and the notable development in the past decade has been the standardization of encryption of DNS queries and responses. Some browsers have been quick to fold encrypted DNS functions into the browser itself, creating a privacy envelope that excludes both the operating system platform and the network from either observing or manipulating DNS queries and responses. Viewed at one level this has the potential to be a clear success for the DNS, effectively countering many forms of covert DNS manipulation. However, it's unclear if this is an unqualified success. Pushing the DNS from an infrastructure function to an application function can encourage a number of fragmentary pressures in the DNS, where each application can place the user into a DNS policy environment of their own devising.

In terms of the larger issue of the so-called "apex predators" in today's surveillance-dominated Internet, DNS privacy has changed nothing. While the DNS could represent a rich vein of information about the user, the existing techniques of user profiling from these large-scale advertising brokers do not rely on access to the DNS, and the IETF's moves to support encryption of the DNS appear to be actively encouraged by these larger enterprises, perhaps as a move to entrench their position and shut out the entry of potential competitors.

Centralization of service in the DNS is a dominant feature of the DNS. There are dominant actors in the name registration business, in the DNS registry business, in the name hosting business and in the name resolution business. At this point in time these are all different enterprises who dominate each sector, but in the ebb and flow of business mergers and acquisitions this may be just a temporary aberration in a larger trajectory to absolute control by a single entity.

Even today this centrality has it's associated vulnerabilities, where an outage of a single provider can cause a very broad outage. The 2016 DYN attack caused large scale outages in the US. The more recent Akamai outage similarly illustrates that level of criticality of these apex DNS actors. The DNS appears to have been swept up into the increasing centralization of content distribution, and recent studies indicate a highly dominate position of Akamai and Cloudflare in DNS hosting, due largely to the way content hosting these days relies on DNS redirection. We are left with an ongoing situation of fewer bigger failures!

What do Users Want from the DNS?

Paul pointed out that at times his DNS service has been hijacked by his security provider for the gain of his security provider, his queries have been filtered by his network provider. His typos have been trapped and redirected to advertisements and he has contributed to a shared DNS history database, all without his express permission or even knowledge in most cases. And his experience parallels the experience of every other Internet user. What users may want, what we have is a far cry from most users' expectations.

In any case, what users actually want is perhaps the key question here. At one level users don't want to be in the situation where this question is even pertinent! Why should we be asked this question at all? Perhaps it is more informative to ask what users assume about the DNS, if they ever think about the DNS at all. They want it to work, to be fast and completely unseen and untouched by them. Most end users never change the settings of their Internet service provider, their operating system platform, and their applications.

Do users want diversity of providers in the DNS? Probably not, although not for the reasons that are commonly offered. Warren Buffett has said that diversification is stupid if you know what you are doing, and a rational choice if you don't. But the vast majority of users would claim no familiarity whatsoever with the DNS, but as we've seen diversity would not help in any case. What users want is a rational default configuration. They would prefer that configuration not to be hostile to their interests, but beyond that they really have no further preference. However, it is not clear that the commercial interests that dominate the DNS has the intention to produce outcomes that reflect even a benign attitude to users. The outcomes we see in the commercial DNS tend towards commercial self-interest that is unremittingly exploitative of users to the exclusion of any other consideration.

If you ask what various DNS providers want, then you would get an entirely different set of response. Depending on their particular role, they would like to observe DNS queries and responses, or selectively alter responses or redirect queries. Some would like to assemble profiles of users, others would like to assemble profiles of malware activity, with others want to intercede in the operation of malware and disrupt it.

All of this is supposedly in the cause of their perception of some form of users' interests or the interests of the network itself. Its challenging to predict where this is heading, and even more challenging to maintain an optimistic outlook that this will be resolved in ways that enhance user privacy and network resilience.

The Curious Court Case of Quad9

Quad9 is an open DNS resolver service, started in 2016 with the support of IBM, the Global Cyber Alliance, and Packet Clearing House. The service is a so-called "clean-feed" service, with block lists assembled from more than 20 security intelligence providers, intended to disrupt various forms of criminal abuse of Internet users through DNS blocking. Quad9 is based in Switzerland and operates under the provisions of the EU GDPR regulations as well as Swiss Data Protection measures. The resolvers are located in some 90 nations, with some 180 points of presence. The service is operated as a free service.

As Quad9's John Todd observed, DNS filtering is an effective way for cooperating users to be protected against certain threats that they wish to avoid. On the other hand, it is entirety counter-productive in attempting to coerce unwilling users in not connecting to certain proscribed services. Many nations have such block lists of proscribed services, and these measures are commonly implemented by service providers as DNS blocks applied to users within that country.

The global open DNS recursive resolver providers pose a challenge to such national measures. The general intention is to provide the same service to all queries irrespective of the supposed location of the querier. Given that most national blocking measures are unclear about who and how such blocks should be maintained, these distributed open resolvers have managed to stay one step removed from such measures. However, IPR interests associated with Sony Music Germany bought a suit against Quad in a German court, that ruled that Quad9 must block resolution of a domain name of a website in the Ukraine that itself does not hold copyright infringing material, but instead contains pointers to another web site that is reported to hold alleged copyright infringements.

Quad9's interpretation of this ruling is that queries from IP addresses that can be geolocated to Germany sent to Quad9 resolver instances located in Germany for queries for this particular domain name will generate a SERVFAIL response. The implication here is that this imposes an additional cost on the DNS service in that it needs to perform a geo lookup and invoke a policy rule in the case of a geo-match.

There are a number of curious aspects of this situation. It appears that the other significant open DNS resolver providers (Google, Cloudflare, and Cisco's OpenDNS) have not been similarly targeted by legal action in Germany by Sony. Perhaps the Swiss domicile of Quad9 made Quad9 a more appealing target for German legal action. Or Quad9's small size made them a vulnerable first target for Sony and the IPR industry. However, it's hard to see the overall rationale for this move in a larger context of geopolitical presence the Internet. We see emerging disquiet in the EU over the dominant position of US corporate interests in almost every aspect of the Internet and the DNS space is no exception. The EU is being largely treated in the same way as the bygone imperial empires treated their colonies, and for the EU it's a novel and deeply discomforting place to be. The EU is trying hard to position EU enterprises in direct competition to these US-based Internet giants. The DNS has been caught up in these efforts and there are some recent EU initiatives, such as the DNS4EU program which is intended to create some EU-based competitive positions. However, this German court decision has the opposite effect. If this makes DNS operators domiciled in Europe more at risk from expensive-to-implement regulatory measures that are not imposed on foreign DNS providers, then this entire EU initiative is probably going to go nowhere useful!

Quad9's preferred response, aside from getting this particular court decision overturned with a legal challenge is to encourage policymakers to specifically mention recursive DNS services as exempt from mandatory censorship requirements. At a minimum they would like to see recursive DNS-based models of content control optional. As far as I can tell, in the words of that immortal Australian film classic, The Castle, "They're dreaming!"

Jio and DNS Tunnelling

Some two thirds of the world's Internet user's direct their DNS queries through ISP-provided recrussive resolvers. This means that some of the larger resolvers are located in some of the larger retail ISPs, and these are located in toe most populous countries. In India the largest retail provider is Reliance Jio, and they have some 430M subscribers. Their DNS queries are directed to one of 23 DNS resolver farms, and they use 265 DNS resolver engines. The peak query rate per resolver is some 300,000 qps and the aggregate peak is some 15.9M qps. It's a large-scale DNS deployment.

They have been asked by the national regulator to block *DNS Tunnels* as these mechanisms bypass the existing national DNS censorship measures. If you want to hide you DNS queries there are two basic options: you can encrypt the DNS packet itself, or you can leave the the DNS packet in the clear and encrypt the query inside the query label. In this case the technique uses the latter option, and it's very similar to the oblivious DNS technique. The query is encoded in Base64 and the new query name is directed to a cooperating decoder, which then performs the resolution on the user's behalf like any normal recursive resolver. The response is encoded again using Base64 and passed back as a TXT response to the original query.

The problem is automated detection of such DNS tunnels is that long seemingly random query names are used in a number of legitimate contexts, including many content data hosting configurations. They believe that they have now deployed an effective tunnel detector and blocker. Interestingly, DNS tunnel traffic was observed to be as high as 2% of the total DNS traffic in their network. If this is an indicator of the level of user demand for bypassing these national blocks, then it's likely that they are now engaged in a somewhat unproductive escalation process of move and countermove with no clear end in sight.

The sheer size of the Jio DNS environment tends to suggest that the advantage lies in the folk devising and promulgating the active bypass techniques in the DNS, while the large-scale deployment of DNS resolvers weighs down the agility of the deployed defensive mechanisms.

Google's Public DNS Service

The largest DNS resolution environment on the Internet is operated by Google. This is their open DNS recursive resolver, that responds to queries passed to 8.8.8.8. This project is now 11 years old (it was launched on the 3rd of December 2009).

Google has been very active in implementing new DNS standards as they are published in a stable form. Their service is now the largest DNSSEC validating DNS resolver system on the Internet, which was launched in March 2013. In recent years Google DNS has introduced support for DNS over TLS (DoT) and DNS-over-HTTPS (DoH) (both in 2019), and in 2020 Google has supported Query Name Minimization (with minimization being performed for up to 3 name levels, as I understand). Google is also using aggressive NSEC caching, cache poisoning protection via 0x20 bit munging, nonce prepending and DNS cookies. They are experimenting with DNS to Authoritative servers with a few selected authoritative operators. Perhaps more controversially, Google supports EDNS Client Subnet. This includes some client information in the query passed onward from the recursive resolver to the authoritative server. It allows the authoritative server to provide a geo-targeted answer based on the assumed location of the client, but at the cost of client privacy and cache efficiency.

The use of Google's service continues to grow, and the number of users passing queries through Google's resolution service has grown by 50% over the past 15 months. Within this there are some other notable trends. The number of queries for IPv6 addresses (AAAA queries) has risen from 2% of queries at the end of 2019 to a current proportion of 7% of all queries (or a little over a tripling of the proportion of queries over this period). This is curious, in that the measurements of the population of IPv6 users have risen from 24% to 30%, or a relative growth of 25%. A possible explanation for this is that the query traffic seen by Google's DNS encompasses more than user queries, and Google's traffic profile may include a sizeable proportion of other query traffic, such as transaction log data analysis and even query log replay. The query volume for DoT and DoH has increased 5-fold, from 2.5% of queries to 12% of queries. The split between DoT and DOH is approximately even these days. This makes a lor of sense for clients of Google's service if its clients are required to reach Google over an Internet path. However, it is not so clear about the context for those ISPs who use Google via a forwarding arrangement (of which there appear to be many).

Google use a multi-tier internet architecture The front-ends are simple caching only resolver engines that handle various encapsulation protocols as well as the DNS protocol with the client. If the query cannot be answered from the local cache the query is passed to a back-end resolver engine which will either answer from its local cache or pass it on for resolution with proxy query servers handling individual queries to authoritative servers. For reasons of simplicity there is no shared cache in this architecture.

For Google there is an increasing level of integration of the DNS into their existing web delivery infrastructure, and the DNS, even without the added impetus of DOH, is being treated in a similar manner to another component of web-based infrastructure with common service elements for TLS termination, HTTP transports, DoS protection and OAM management. This allows the Google DNS service to leverage existing Google service modules where available. For example, the DNS service uses a larger Google DoS blocking service and does not need to separately operate its own DoS detection and blocking service.

The system is intended to operate in a largely autonomous manner, which, with the exception of very large DoS attacks and outages of certain TLDs, has been successful so far. The DoS defence mechanisms are tuned to detect certain forms of query attacks and not pass these queries on to the authoritative servers. The system also has some rate limits so as to limit the capacity of reflection attacks that attempt to use Google's DNS.

At this point Google maintain that they perform no DNS filtering or censorship of any form in any of their points of presence. Obviously, the issues behind the German court decision relating to Quasd9 are not going away any time soon.

Summary

The DNS appears to operate in a style that is like other collaborative group projects, in that the group appears to pick up a theme and run with it intensively for a while and then apparently lose interest and move on to the next theme.

For a while the range of defensives techniques to use against DNS-based DoS attacks were a consistent theme of DNS conversations. We then moved on to the issues of the increasing baroque ornamentation of the DNS protocol in the "DNS Camel" conversations. The past couple of years has seen what could be called an obsessive interest in channel encryption for the DNS. More recently we've been looking at the service records, and SVCB and HTTPS records, and ways that the DNS can be augmented to provide an application-level rendezvous function, complete with a set of application-level protocol parameters, as distinct from basic name-to-address mapping.

There is a larger tension at play as well between scalability and functionality of DNS responses. The initial efforts in the DNS were directed to provide consistency and uniformity of responses. The positive benefit of this was caching, which in turn was a major reason why the DNS was able to scale so readily. Caching pushed DNS traffic out to the edge of the network, which reduced the traffic impacts on authoritative servers. This, in turn, has allowed individual zones to bloat in size, most notably the .com and .net zones, where .com now contains more than 145M name registrations and .net contains a little over 13M names.

At the same time, we have been centralizing increasing amounts of DNS infrastructure. Within the ISP the DNS recursive function is being operated within the ISP by outsourced providers, such as Secure64 or Akamai's AnswerX. This is complemented by the rise of a small set of open DNS resolvers, notably Google's DNS service. Over on the authoritative server size there is similar centralization, where a small number of large enterprises operate much of the DNS' server infrastructure, including Amazon's Route53, Akamai's Edge DNS and Google's Cloud DNS.

This means that we are more confident of the DNS infrastructure to scale to ever larger levels. In terms of the tension between scalability and customization within the DNS we are less concerned about scalability. These days we are confident in running very popular domain names with short TTLs (Facebook's outage was triggered by short TTLs) and using the DNS to perform content steering and rendezvous (such as can be achieved by a combination of EDNS Client Subnet and HTTPS records). We are evidently prepared to reduce the levels of caching and place greater stress on the DNS as a result. The use of Chrome's sensing queries and the use of nonces in discovery queries as a means to prevent certain forms of DNS poisoning also deliberately bypass local DNS caches.

It's not just downplaying the benefits of caching. We are now also prepared to contemplate ditching UDP and not only heading to DNS over TCP, but to also introduce encryption into the process. It was long believed that the use of UDP was not just a simplification for the DNS, but a core element of the efficiency of the DNS protocol.

These days the tension between scalability and functionality in the DNS is currently favoring functionality at the expense of cache effectiveness and scalability. How long we can sustain this stance and allow the DNS to continue to chase down various forms of functional bloat is, at this stage, anyone's guess!

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