June 2024
Geoff Huston

DNS Evolution

The DNS is a crucial part of today's Internet. With the fracturing of the network's address space as a byproduct of IPv4 address run down and the protracted IPv6 transition the Internet's name space is now the defining attribute of the Internet that makes it one network. However, the DNS is not a rigid and unchanging technology. It has changed considerably over the lifetime of the Internet and here I'd like to look at what's changed and what's remained the same.

The Early DNS

The early Internet architecture used names as a convenient alias for an IP address. Each host used a local list of name and address pairs, and an application would look up the name in this file and use the associated address to use in the subsequent packet exchange. In many ways this file was a direct analogy to the telephone directory in a telephone network.

This simple framework has one major drawback, namely scalability. As the number of connected hosts on the network increased, the burden of distributing updated copies of this file increased and the task of maintaining loose coherence across all these local copies of the name file became more challenging. The document IEN 61, describing an Internet Name Server, was released in 1978, and it appears to be a very basic predecessor of today's DNS.

Some five years later, in 1983, RFC 882 defined a hierarchical name space, using a tree-structure name hierarchy. It also defined a name server as a service that holds information about a part of the name hierarchy, and also referrals to other name servers that hold information about lower parts of the name hierarchy. The document also defined a resolver which is able to resolve names into their stored attributes by following referrals to find the appropriate name server to query, and then obtaining this information from the server. RFC 883 defined the DNS query and response protocol, a simple stateless protocol.

And that's about it.

Most of what is in today's DNS was defined in these early specifications, and what we've been doing over the intervening forty years been filling in the details. The DNS not really changed in any substantive manner over the intervening period.

Evolutionary Pressures

However, I think that such a perspective ignores a large body of refinement in the DNS world that has occurred.

The DNS is by no means perfect. The DNS can be extremely slow to resolve a name, and even slower to incorporate changes into the distributed data framework. The resolution of DNS queries pays scant regard to concerns about user privacy, and any party who can observe a user's DNS query stream can readily piece together an accurate picture of the user's activities. The distributed stateless method used to resolve names is prone to various efforts to eavesdrop DNS transactions and manipulate the information being provided in DNS responses. The DNS cannot easily protect itself from disruptive attack and has

been regularly used in highly effective denial of service attacks. It's also insecure, in that the authenticity and currency of a response cannot be verified by a client.

The operation of the DNS in resolving a name can be extremely opaque. The use of parallel servers and resolvers to improve the resilience of the DNS creates combinatorial explosion in the number of paths that can be used to navigate through the distributed data structure. It is not possible to tell in advance which servers may be used in the resolution of a query, or the number of additional queries a single original query may trigger. Given that resolvers are able to respond directly a query from a locally cached response, it is not possible to tell in advance where the response came from, or if the response is authentic.

For a common and fundamental service that every user not only uses, but implicitly relies upon, the DNS in practice is far from a paragon of sound operational engineering.

The evolutionary efforts have been intended to remedy some of these shortcomings, intending to improve the speed of DNS resolution, improving aspects of privacy of DNS transactions, increasing the level of trust in DNS responses, and resisting efforts to subvert the integrity of DNS name resolution transactions.

DNS Privacy

The DNS is not what you might call a discrete protocol. By default queries are made in the clear. The IP address of the querier, the server being queried and the name being queried is visible to any party that is in a position to inspect DNS traffic. This not only includes potential eavesdroppers in the network, but the operating system platform that hosts the application making the DNS query, the recursive resolver that receives the query, and any forwarding agent used by the recursive resolver. Depending on the state of the local cache in the recursive resolver, the recursive resolver may need to perform some level of top-down navigation through the name server hierarchy, asking an authoritative server at each level the full original query name. The recursive resolver will normally list itself as the source of these queries, so the identity of the original user is occluded, but the query name is still visible.

RFC 7858 provides a specification for DNS over a Transport Layer Security (TLS) session (DOT). This allows the client and server to set up a shared session key in a secure manner which is then used to encrypt all subsequent transactions between the two parties. TLS can also be used to authenticate the server name, to assure the client that it is connecting to an instance of the named server. There is some overhead to setting up a TLS session, and the most efficient use of this approach is in the stub-to-recursive DNS environment where a single TLS session can be kept open and reused for subsequent queries, amortizing the initial setup overheads across these queries. The standard specification of DOT defines the use of TCP port 853, which allows an onlooker to identify that DOT is being used and identify the two end parties by their IP addresses, but not the DNS queries and responses.

Subsequent standards work has defined DNS over QUIC, RFC 9250 (DoQ). The encryption provided by QUIC has similar properties to those provided by TLS, while QUIC transport eliminates the head-of-line blocking issues inherent with TCP and provides more efficient packet-loss recovery than UDP.

In addition, it is possible to add an HTTP wrapper to the DNS data object, defining DNS over HTTPS RFC 8484) (DoH). DoH uses port 443, using either TCP in the case of HTTP/2 or UDP with the QUIC-based HTTP/3, so the DNS transactions would be largely indistinguishable from Web traffic. HTTP adds its own ability to perform object caching, redirection, proxying, authentication, and compression beyond that provided in the conventional DNS model, although the use of such HTTP capabilities in the DNS context is not well understood. HTTP also allows a server to push content to a client. In the DoH scenario this could permit the use of queryless DNS, where the server pushes DNS responses to a client without any initial triggering DNS query.

In these approaches to encrypted transport for the DNS the remote server is aware of the client's IP address and the queries that the client is making. In the stub-to-recursive scenario this allows the recursive resolver to be privy to the user's DNS actions, even when the network path between the two parties is secure. A stronger level of privacy is obtained by the use of Oblivious DNS over HTTPS (RFC 9230) where no single DNS server is simultaneously aware of the client's IP address and the content of the DNS queries. Here a double level of encryption is used in conjunction with two independent agents within the network. The client sends an encrypted DNS query to the first proxy, using DoH. This proxy is aware of the client's IP identity, but is not able to decrypt the DNS query. The proxy makes its own query using the encrypted query to a separate target, again using DoH, but this time there is no record of the original client. The target can decrypt the query and function as a conventional recursive resolver

These four specifications show that it is possible to cloak DNS transactions within a secure veil of secrecy, but it remains a topic of speculation as to the extent of uptake of these technologies. Encrypted transport sessions impose higher costs on the operation of DNS infrastructure (recursive resolvers and authoritative servers), and it is unclear how these higher costs can be absorbed into the current DNS economic models where individual DNS queries are essentially unfunded by the client.

An entirely different approach to improving DNS privacy is described in DNS Query Name Minimisation (RFC 7816). The observation is that as a recursive resolver navigates its path through the DNS hierarchy it uses the original query name to query authoritative name servers, essentially sharing the knowledge of the name being queried with a set of servers. The rationale for this approach is that the client does not necessarily know where a zone cut may exist in advance. Query Name minimisation proposes to minimise the amount of information being disclosed to authoritative name servers by sending a request to the name server authoritative for the closest known ancestor of the original query name, and asking for a delegation record (NS) rather than the original query type. This approach does not impose additional overheads on DNS server infrastructure. It does not offer channel security, but it does limit the amount of information 'leakage' that is a feature of the DNS name resolution process.

On a more general level, none of these DNS privacy measures can assure a user of the authenticity of the DNS response that they receive. These measures limit the ability of other parties to eavesdrop on DNS queries and responses, but detecting (and presumably rejecting) DNS responses that are inauthentic in a separate issue for the DNS.

DNS Authenticity – DNSSEC

DNSSEC is an extension to the DNS that associates a cryptographically generated digital signature with each record in a DNSSEC-signed zone, specified in RFC 4033. DNSSEC does not change the DNS name space, nor the DNS name resolution protocol. Clients who are aware of DNSSEC can request that a DNS response should include a DNSSEC signature, if one is available for the zone, and may then validate the response using the signature.

You might think that a tool that allows the client to verify a DNS response would be immediately popular. If the relationship between the names that are used by applications and services and IP addresses that are used at the protocol level is disrupted, then users can be readily deceived. Yet, after close to three decades after its initial specification DNSSEC is still struggling to achieve mainstream adoption. Part of the issue is the strong binding of the DNS protocol to a UDP transport cause a set of issues when responses bloat in size due to attached signatures and keys. Part of the issue lies in the care and attention required to manage cryptographic keys and the unforgiving nature of cryptographic validation. And a large part of the issue is when the web took to using TLS as a means of verifying the identity of a remote server, then any marginal incremental benefit of DNSSEC in the DNS part of session creation was considered to be not worth the incremental effort and cost of using DNSSEC.

For these reasons DNSSEC continues in the DNS environment as a "work in progress".

Evolution of Query Mechanisms

The base DNS specification uses a limited repertoire, where queries contain a query name and a query type, and DNS responses, if carried over UDP, are limited to 512 byes in length. The restrictions in the size of several flag fields, return codes and label types available in the basic DNS protocol was hindering the development of DNSSEC. The chosen path to resolve this was the use of a so-called pseudo Resource Record, the OPT record, that is included in the additional data section of a DNS message. To ensure backward compatibility a responder does not use the OPT record unless it was present in the query. This is the general extension mechanism for DNS, or EDNS.

EDNS options have been used so far to support DNSSEC functions, padding, TCP keepalive settings and Client Subnet fields. It has also been used to extend the maximum size of UDP messages in the DNS through the use of a EDNS Buffer Size.

It is often desirable to separate the name of a service and the location of the service platform that delivers the service, and service record type was intended to achieve that outcome. Service Records, SRV records, can provide that form of flexibility, where the service is defined by a host name, a port identifier and a protocol identifier, and the associated resource record provides the TCP or UDP port number and the canonical service name of the target service platform. Multiple service targets can be specified with an associated preference for use. The functional shift in the use of the SRV record was loading the DNS query with a service profile rather than a plain domain name, and in return receiving enough information to enable the user to then connect to the desired service without making further DNS queries.

This was further extended in the SVCB specification (RFC 9460). By providing more information to the client before it attempts to establish a connection, these records offer potential benefits to both performance and privacy. This represents a shift in the design approach of the DNS, where the prior use of DNS resource record types was to segment the information associated with a DNS name, so that a complete collection of information about a service name was obtained by making a set of queries. The SVCB record effectively provides a "omnibus" response to a service query, so that the client is able to gather sufficient information to connect to a service with a single DNS transaction.

Delegation Records

One of the fundamental parts of the DNS data structure is the delegation record, which passes the control of an entire subtree in the DNS hierarchy from one node to another.

While this NS record has served the DNS since its inception, it has a few limitations. The target of the delegation record is one or more DNS server names, not their IP addresses. Conventionally the IP addresses are provided as "glue records" contained in the Additional Section of a DNS referral response, but the veracity of such glue records cannot be established, and has been the focal point of a number of DNS attacks over the years. The target of a NS record cannot be a CNAME alias. The NS record is shared across bother the parent and the child zones, and the child zone is deemed to be authoritative for this record. This implies that while the parent zone name servers can (and must) respond with referral responses with this NS record, it cannot provide a DNSSEC-signed response. It is not possible to provide a DNS service profile in a referral response. If the zone's authoritative servers can be accessed using an encrypted transport protocol, this capability cannot be signalled by the NS record.

There is work underway in the IETF in the Deleg Working Group to take the existing specification of service binding mapping for DNS servers (RFC 9461) and see how this could be used as a more flexible delegation record that addresses some or all pof these identified shortcomings in the existing NS form of delegation.

Alternate Name Systems

The Internet protocol suite can be regarded as a collection of elements, including addressing, routing, forwarding and naming, and it's possible to substitute a different technology for one element without necessarily impacting on the others. For example, the transition from IP version 4 to IP version in the

addressing realm does not necessitate any fundamental changes to routing, forwarding or naming. The same can be said of the DNS name system. Alternate name systems can be used and to some extent they can coexist with the DNS.

In the traditional model of DNS resolution, users have little control over their DNS settings. Some technically literate users may choose settings that differ from the defaults, but there has been little incentive to do so, and the vast majority of users have their DNS settings configured for them by administrators via a protocol such as DHCP.

Many alternative naming systems in use today come bundled with the specific applications that use them: a particular alternative naming system is often tied to a corresponding application, and this application often bypasses administrator-controlled settings and any pre-configured DNS settings. For example, the Tor Project uses its own naming system that bypasses traditional DNS resolution. A user can install the Tor Browser, and it will use the Tor naming system for names ending in .ONION, while forwarding any other names to the local DNS library. The application developer makes the choice of which naming system to use without the user even knowing that they are using an alternative naming system nor understanding potential implications.

Versions forms of experimentation have used decentralised models which eschew a single name hierarchy and allows individual names to exist in an unstructured flat namespace. The underlying registry framework that associates a name with an "owner" has often relied on some blockchain-like approach, where the association of a name and a public key value is placed into the blockchain. A number of such alternate name systems exist today, including the Etherium Name service's ENS, which makes use of so-called "smart contracts" in its blockchain, Unstoppable Domains which uses a blockchain platform but operates the name space as a centrally operated space. The GNU Name System (GNS) is also a decentralised platform that offers name persistence. GNS has no concept of a root zone. Instead GNS uses the concept of a "start zone" that is configured locally and determines where to begin resolution. Since local users have complete control over their own start zone, every GNS user can potentially use a different namespace. Thus, there is no guarantee that names will be globally unique, or that a given name will resolve the same for different users. The only guarantee is that users with the same start zone will have the same view of the namespace. Every unique start zone defines its own namespace. This is similar in practice to DNS resolution using different root zones. The key innovation in GNS is to replace a search hierarchy with a distributed hash table that can include links to other hash tables.

Such alternate name systems interact with the existing DNS-defined name space in a variety of ways. Some attempt to coexist with the DNS, and the alternate names being some form of extension to the DNS name space, potentially associated with a different name resolution protocol. Other systems are completely self-contained and make no effort to coexist with the DNS. This is more commonly seen in an application-specific context where the application environment is exclusively associated with an alternate name space.

Conclusions

Only a completely moribund technology is impervious to change! As digital technologies and services evolve, the demands placed on the associated namespaces also evolve in novel and unpredictable ways.

The DNS is an interesting case in point that so far it has been able to respond to the evolving Internet without requiring fundamental changes to the structure of its name space, to the distributed information model nor to the name resolution protocol. Most of the evolutionary changes that have been folded into the DNS to date have been undertaken in a way that preserves backward compatibility, and th4e cohesion of the underlying name space has been largely preserved.

However, maintaining this cohesion across the Internet is not an assured outcome looking into the future. The pressures at a national and regional level to impose barriers to the access to content are often expressed in the manner of imposing selective barriers to the resolution of content service names, and

the DNS is left carrying the burden of supporting such selective fragmentation in the Internet. The camel has undeniably poked its nose into the tent of name coherence in the form of EDNS Client Subnet, where the response given to a query may be dependent on who is querying, as much as the name that is being used in the query, and it's likely that this more qualified and fragmented model of a name space will persist and support an increasingly fragmented Internet.

Disclaimer

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

Author

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

www.potaroo.net