The ISP Column
A monthly column on things Internet

October 2016

Geoff Huston

IPv6 and the DNS

The exhortations about the Internet's prolonged transition to version 6 of the Internet Protocol continue, although after some two decades the intensity of the rhetoric has faded and, possibly surprisingly, it has been replaced by action in some notable parts of the Internet. But how do we know there is action? How can we tell whether, and where, IPv6 is being deployed in today's Internet?

A number of groups, including the APNIC Labs group, perform extended measurements of the Internet, and report on the current state of deployment of IPv6. As a result, we can now state, with some authority, that there is large scale IPv6 deployment by service providers in Belgium, the United States, Switzerland, Greece, Germany and Portugal. We can take these per-country IPv6 measurement and use it to color a map of the world (Figure 1).

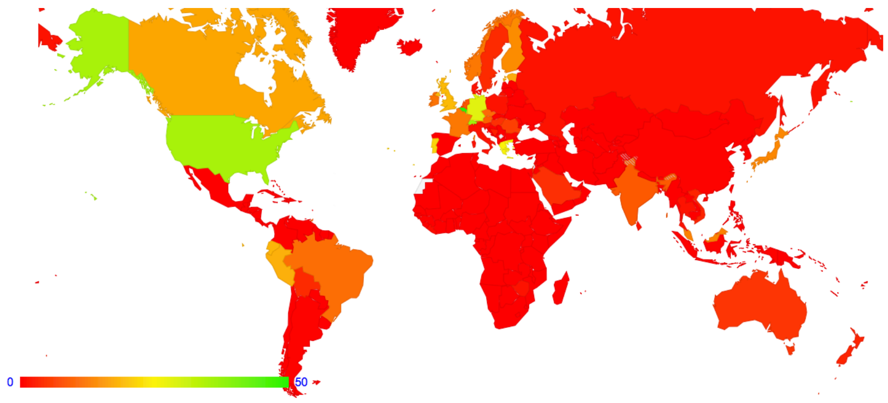


Figure 1 – Global IPv6 deployment, as measured by the relative capability to use IPv6 per country (http://stats.labs.apnic.net/IPv6)

We can also take this data and produce a time series of IPv6 deployment for the entire Internet, showing progress over time (Figure 2).

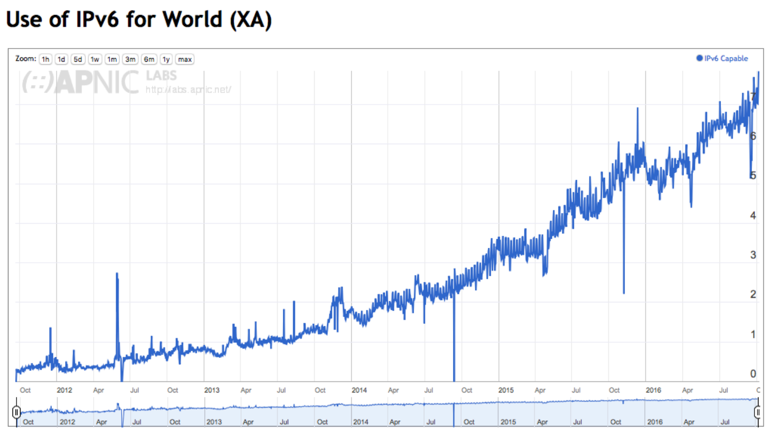


Figure 2 – Global IPv6 deployment as a time series (http://stats.labs.apnic.net/IPv6/XA)

Figure 2 shows us that the global average level of IPv6 deployment is now around some 7%. What does that number mean? This measurement is indicating that the estimate of the proportion of the Internet's user population that can retrieve web object across IPv6 is some 7% of the total user base. In other words, if you hosted a web site and made it accessible only using IPv6, then some 7% of the total user population would be able to access the service, in the hypothetical situation where everyone on the Internet decided to access this IPv6 web object.

However, this is perhaps not quite the entire story about the transition to IPv6. To access a service a user would conventionally use the DNS to resolve a name into an IPv6 address. Other services are also part of the Internet environment, such as network time, electronic mail, secure transport services, and so on. While the IPv6 capability of web object retrieval is a reasonable measurement to perform, it's not the complete picture. One of the more critical of these 'hidden' services is the Domain Name resolution protocol, which I'll refer to here as the DNS.

How are we doing with IPv6 support in the DNS?

The DNS is a multi-faceted environment, populated by publishers of information (authoritative services), caching middleware (in the form of recursive resolvers) and query agents. So what aspect of IPv6 and the DNS are we talking about? The way in which clients pass queries to resolvers? Or the way in which resolvers pose queries to authoritative name servers? There is also the distinction between whether the query is made using IPv6 or IPv4 as the IP substrate for the DNS transaction, or whether the DNS query itself is about attempting resolve a certain domain name to an IPv6 or IPv4 address. There is also the aspect of published DNS data, and counting the extent to which domain names are published with both A (IPv4) and AAAA (IPv6) address records.

Each of these questions probably require a different form of measurement, and probably lead to differing answers. Here we will look at just one question, and report on that. The question is:

How much of the DNS resolution infrastructure is IPv6 capable?

This is a question about the use of IPv6 as the substrate for DNS queries.

However, even then it's a deceptively challenging question to attempt to answer.

The DNS may look like a straightforward query/response protocol, but in the infrastructure of the DNS there is a lot going on. End users are often configured with two or more resolvers. That means that users' DNS queries are often directed to different recursive resolvers. Most DNS resolution libraries try to make use of this local diversity by duplicating queries to the other resolvers if no answer is forthcoming from the initial query. Recursive resolvers may pass their queries authoritative name servers, or they may, in turn, forward their queries to another recursive resolver. They may take the latter option as a means of concentrating queries, which in turn has the potential to increase the caching effectiveness of the target recursive resolver which may, in turn, improve the DNS resolution times for the user. Large recursive resolvers may form part of a server "farm" as a means of scaling capability. Making this more challenging is that the DNS protocol does not necessarily write a referral trail into each query. There is no timestamp, no query time to live, or record of the original agent who initiated the query. It is also misleading to speak of a single "query" in such a context. A request to resolve a DNS name may initiate a search within the hierarchical DNS name space, so that one original query may trigger a sequence of queries.

If we take the perspective of an authoritative name server, the DNS starts to look a lot like a fuzzy 'cloud' of semi-opaque behaviour (Figure 3). A user may have initiated a DNS transaction to resolve a name to an address, and the result is that one or more resolvers pose queries to the authoritative name server asking for the server's information about this name (Figure 3). We will call these resolvers that are seen to pose queries to authoritative name servers "visible" resolvers. (From the perspective of the authoritative name server, the resolvers used by the client, and any intermediary forwarding resovlers are essentially hidden from the server.)

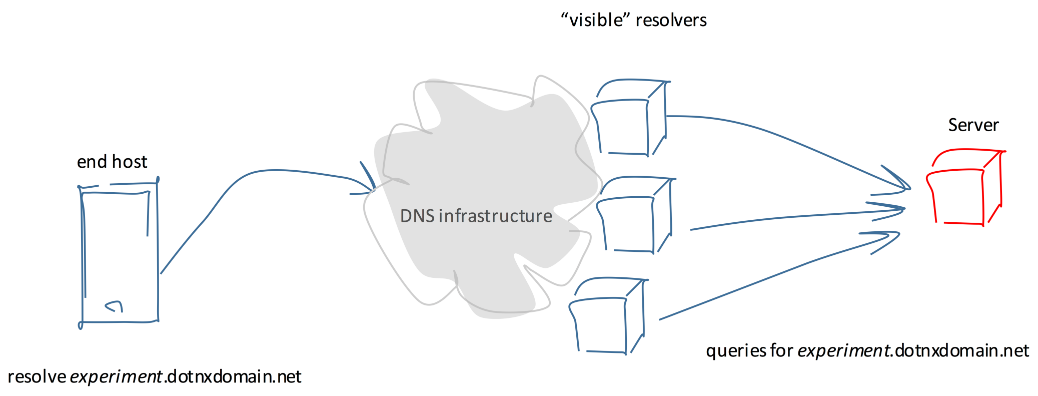

"visible" resolvers
Server
end host
DNS infrastructure
queries for *experiment*.dotnxdomain.net
resolve *experiment*.dotnxdomain.net

Figure 3 – A Conceptual Framework for DNS Resolution

In attempting to answer our question, we'll ask a slightly simplified version of the question: What proportion of "visible" resolvers are capable of posing a DNS query over IPv6, and what proportion of end users are served by these IPv6-capable visible resolvers?

To answer this question, we use an online ad distribution network to feed a small set of URLs to end users. The DNS name is uniquely served by an authoritative name server, so that any queries that are made the authoritative name server are collected by these servers. To ensure that we get as complete a collection as possible of queries each DNS name contains a unique label component. That way the intermediate resolvers' caches are bypassed when the name is first queried. The name also contains an encoded version of the time, to allow the subsequent analysis to disambiguate between current queries and query replay.

In this way we can tell if a visible resolver asks a query, but can we tell if this resolver has received the answer? Given that these DNS queries are a precursor to a URL fetch, then if the name resolution is successful then we will probably observe a subsequent fetch to the web server for the named URL. What this web fetch implies is that at least one of the visible resolvers that queried for the name successfully received the DNS response, but in the case where two or more resolvers performed the query, the result is still unclear in terms of which resolver has a capability that is demonstrated not just by posing the query, but also by receiving the answer.

There is at least one way to expose the behaviour of each resolver, even when two or more resolvers are performing queries for the same domain name. The approach uses the delegation process and the so-called "glue" records in a zone file. When a name is delegated, the authoritative parent zone normally includes the IP address of the delegated zone's name servers as additional information. This assists the resolver in following the delegation chain. These "glue" records can be found everywhere. Here's a snippet from the root zone file:

```
bugatti.            172800  IN      NS      a0.nic.bugatti.
bugatti.            172800  IN      NS      a2.nic.bugatti.
bugatti.            172800  IN      NS      b0.nic.bugatti.
bugatti.            172800  IN      NS      c0.nic.bugatti.

a0.nic.bugatti.     172800  IN      A       65.22.208.9
a0.nic.bugatti.     172800  IN      AAAA    2a01:8840:ca:0:0:0:0:9
a2.nic.bugatti.     172800  IN      A       65.22.211.9
a2.nic.bugatti.     172800  IN      AAAA    2a01:8840:cd:0:0:0:0:9
b0.nic.bugatti.     172800  IN      A       65.22.209.9
b0.nic.bugatti.     172800  IN      AAAA    2a01:8840:cb:0:0:0:0:9
c0.nic.bugatti.     172800  IN      A       65.22.210.9
c0.nic.bugatti.     172800  IN      AAAA    2a01:8840:cc:0:0:0:0:9
```

Figure 4 – A snippet of the DNS Root Zone showing the use of glue records

If a resolver was attempting to resolve the name `aaa.example.bugatti` then an initial query directed towards the root servers would return in the additional section of the response this list of name servers and their IP addresses, allowing the resolver to direct its next query to an authoritative name server for the `bugatti` zone, and so on. While these glue records are useful, they are not strictly required. If a resolver performing a name resolution encounters a delegation without accompanying glue records it must pause the primary resolution task and commence resolution of the names of the name servers where there was no glue. Once this completes it can now resume its primary task.

We can use this behaviour to isolate particular capabilities of each visible resolver. Figure 5 shows a set of zone snippets that have a glueless delegation.

zone dotnxdomain.net

```
experiment   IN NS   srv1.ns.nxdomain.net.
```

zone nxdomain.net

```
ns                   IN NS srv0.ns.nxdomain.net.
srv0.ns.nxdomain.net IN A 192.0.2.2
```

zone experiment.dotnxdomain.net

```
abc    IN A 192.0.2.1
```

zone ns.nxdomain.net

```
srv0 IN A 192.0.2.2
srv1 IN A 192.0.2.3
```

Figure 5 – Glueless delegation example

In order to resolve the name `abc.experiment.dotnxdomain.net` the resolver will query the name server for `dotnxdomain.net` and receive a response that indicates that the name `experiment.dotnxdomain.net` is a delegated domain and the name server of this delegated domain is `srv1.ns.nxdomain.net.`, without any additional information providing the server's IP address (as the delegation is "glueless").

At this point the resolver now has to resolve this name server name, and to do so the resolver will perform a second top down sequence of queries to resolve this name server name. In doing this, the resolver will query the `.net` server for the name `srv1.ns.nxdomain.net` and the response will be the name servers for `ns.nxdomain.net` and their IP addresses (the "glue"). Using this glue record the resolver will query this address (the IP address of the server for `ns.nxdomain.net)` for the address of `srv1.ns.nxdomain.net.` At this point the resolver has now completed its secondary task, and is now in a position to resume its original task. The resolver can now query the server `srv1.ns.nxdomain.net` for `experiment.dotnxdomain.net` for the name `abc.experiment.dotnxdomain.net`.

How can we use this approach to expose a resolver's IPv6 capability?

In the above example we can change the address of the name `srv0.ns.nxdomain.net` to an IPv6 address (Figure 6).

zone dotnxdomain.net

```
experiment   IN NS   srv1.ns.nxdomain.net.
```

zone nxdomain.net

```
ns                   IN NS srv0.ns.nxdomain.net.
srv0.ns.nxdomain.net IN AAAA 2001:db8::1
```

zone experiment.dotnxdomain.net

```
abc    IN A 192.0.2.1
```

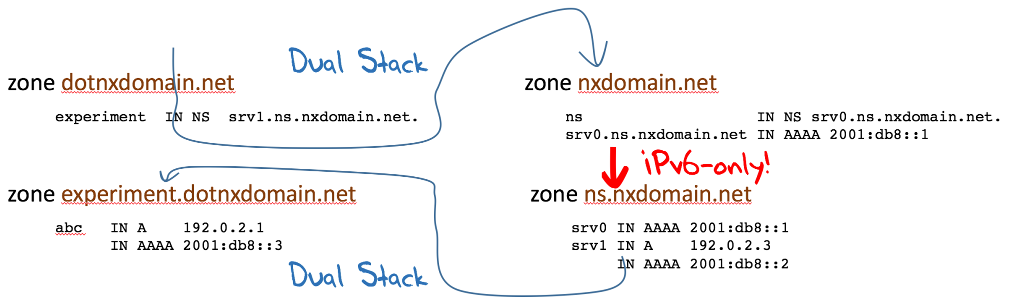
zone ns.nxdomain.net

```
srv0 IN AAAA 2001:db8::1
srv1 IN A    192.0.2.3
```

Figure 6 – IPv6 Glueless delegation example

In this example the resolver can only complete the resolution of the name `srv1.ns.nxdomain.net` if it can perform a query using IPv6. This is because the only name server record for the zone `ns.nxdomain.net` only has an IPv6 address. Unless the resolver can generate a query using IPv6 as a DNS transport then the entire resolution process comes to a halt and no further progress can be made (Figure 7).

The authoritative server will see the query and generate the response, and of course this DNS transaction is generally performed over UDP. This means that we are unable to directly confirm if the resolver received the IPv6 response. One way that we can confirm that the resolver received the response is if the same resolver subsequently makes a query for the original name, `abc.experiment.dotnxdomain.net,` to the server `srv1.ns.nxdomain.net (whose IP` address was in the payload of the previous IPv6 response.)



```
zone dotnxdomain.net
    experiment  IN NS  srv1.ns.nxdomain.net.

zone experiment.dotnxdomain.net
    abc    IN A     192.0.2.1
           IN AAAA 2001:db8::3

zone nxdomain.net
    ns                    IN NS srv0.ns.nxdomain.net.
    srv0.ns.nxdomain.net IN AAAA 2001:db8::1

zone ns.nxdomain.net
    srv0  IN AAAA 2001:db8::1
    srv1  IN A     192.0.2.3
          IN AAAA 2001:db8::2
```

Figure 7 – IPv6 Glueless delegation resolver path

This is the basis of the measurement experiment we are reporting here. (The are some details about the use of dynamically generated names that permits the experiment to circumvent the resolvers' conventional caching behaviour, but we won't describe that mechanism here. Suffice it to say that the experiment names are dynamically generated, and are sufficiently unique to ensure that we are not losing queries to these caches.)

We used this DNS configuration in a measurement experiment across July, August and September 2016, serving between 5M and 10M experiments per day through an online advertisement campaign, resulting in some 400 million individual experiments.

We start with the collection of resolvers that query the "parent" server. If they query the "sibling" name server using IPv6 then we can conclude that the resolver appears to be IPv6 capable. If they then query the "child" server then we can conclude that the received the IPv6 response.

Results

Over this period, we observed some 345,394 unique resolvers that asked the authoritative server for the "parent" zone. The "glueless" answer directed these resolvers to resolve a name that has a IPv6-only step. Of these 345,394 resolvers, some 17,804 resolvers were using IPv6 to query the parent zone server, while the remainder had to switch protocol to resolve the name server record, assuming of course that the resolver is dual-stack capable.

Of the remaining 327,590 unique IPv4 resolver addresses, the overall majority of these resolvers did not pose a query to the IPv6-only zone. We only saw a further 1,212 IPv6 resolver addresses making queries to this zone.

Some 268,218 unique IPv4 resolver IP addresses queried the initial dual stack parent domain, but did not query the target "child" domain, leading to the supposition that these resolvers operate as IPv4-only resolvers. The remaining 59,372 IPv4 resolvers that queried the parent zone using IPv4 appear to have some IPv6 counterpart, and operate as some form of dual stack resolver. i.e. some 18% of the set of visible IPv4 resolvers seen in this experiment are capable of using IPv6 to make DNS queries.

However, this result is not necessarily representative of either what end users might expect, nor does it necessarily reflect the anticipated distribution of query traffic on a dual stack authoritative name server.

The distribution of users to visible resolvers is highly skewed, with the overall majority of users sending their queries to a small number of visible resolvers. The distribution of use of these visible resolvers is shown in Figures 8 and 9.

Figure 8 shows the distribution of use of resolvers across all of the 345,394 visible resolvers. Clearly, a far smaller pool of resolvers is used by 90% of the users who ran this experiment. This is shown in Figure 9.

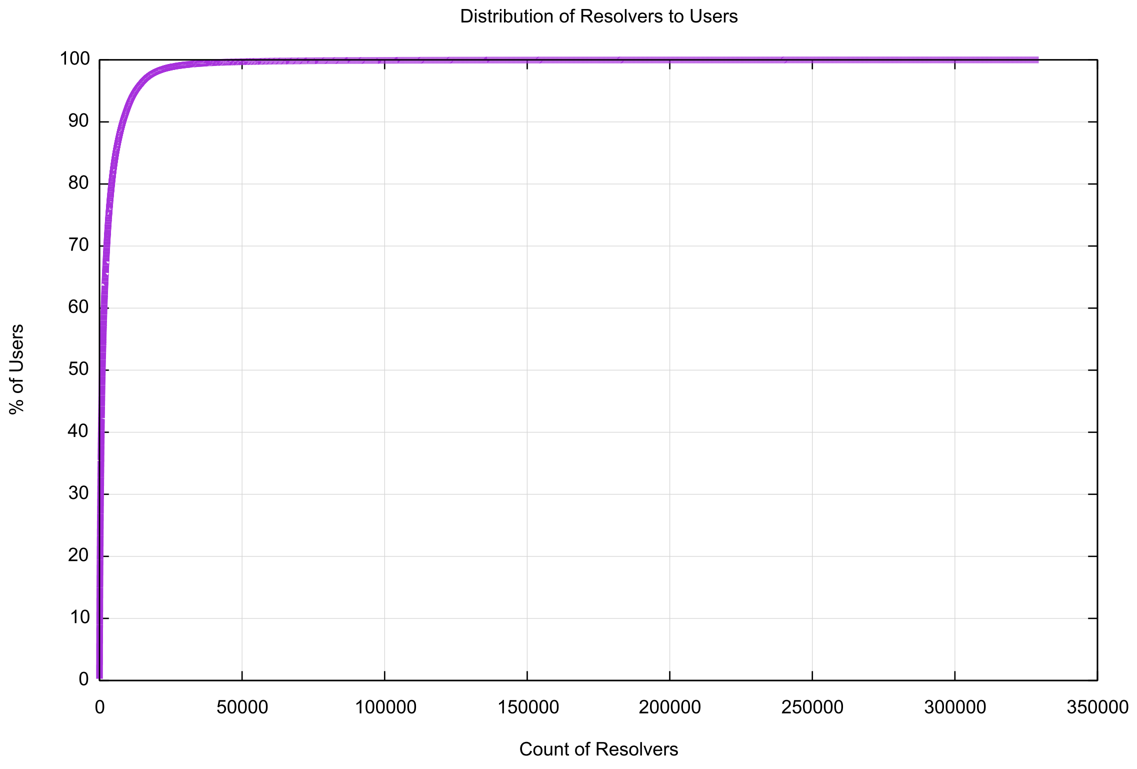


Figure 8 – Distribution of use of visible resolvers – all experiments

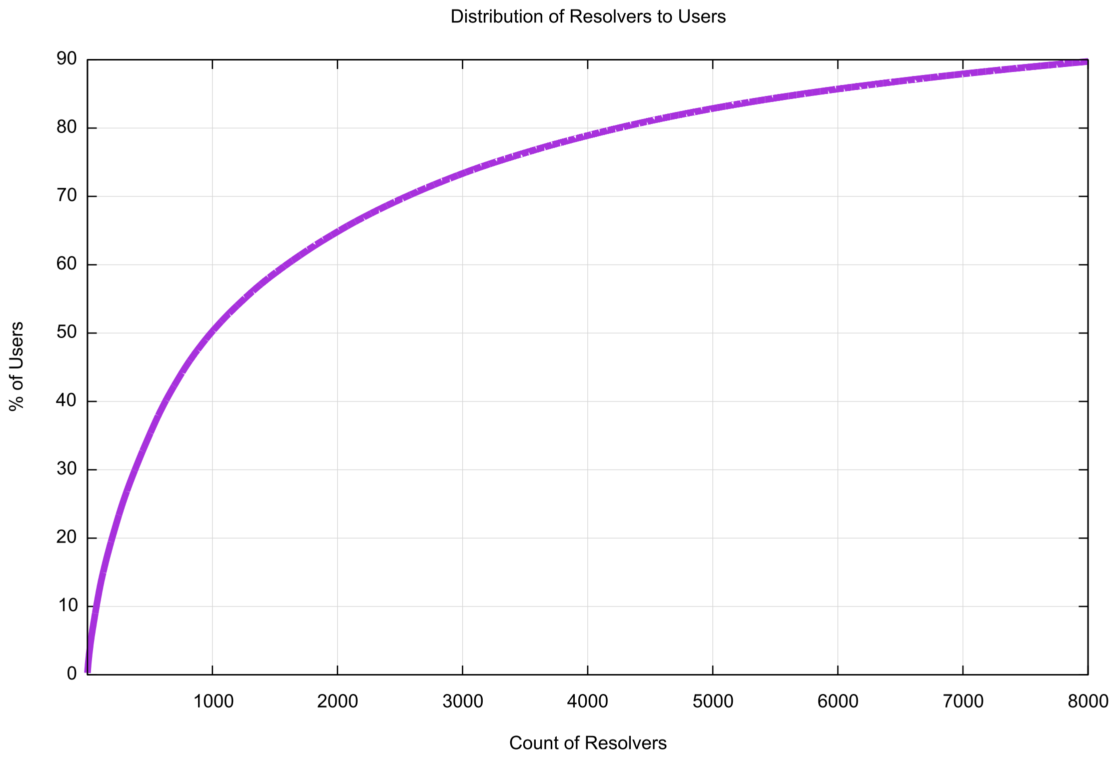


Figure 9 – Distribution of use of visible resolvers – 90% of experiments

It is evident that some 8,000 visible resolvers (by IP address) handle 90% of the users' experiment queries. (It should be noted that the number of actual resolvers handling this load is smaller than this 8,000 number, as the number of unique IP addresses querying authoritative name servers does not take into account DNS resolver farms, such as that operated by Google in their Public DNS service, or by other large multi-headed resolver instances).

Rather than looking at the results by visible resolver, we can look at the results by usage:

Query Sequences	**Parent Only**	**Parent + Sibling**	**Parent + Sibling + Child**
194,137,877	122,667,599	2,970,915	68,499,363
	63%	1.5%	35%

There were 194 million individual query sequences (where a "query sequence" is a unique query string) observed at the authoritative name servers for the "parent" zone. Of these query sequences, some 123 million stopped at the parent and did not perform the IPv6 query of the sibling domain, presumably because they were using an IPv4-only resolver. The remaining 71.4 million query sequences asked the "sibling" server for the NS zone using IPv6. Of these sequences, some 68.5 million of these query sequences asked the "child" server for the target name. This implies that some 35% of all the query sequences were able to successfully use IPv6 to resolve a DNS name. The drop of some 2.9 million query sequences between asking the IPv6 "sibling" query and asking the consequent "child" query (equating to a drop rate of some 1.5%) points to some lingering issues in the reliability of end-to-end IPv6 datagram services.

From these figures it appears that 35% of users, or one third of the Internet's user population, invoke DNS resolvers that are capable of using IPv6 to resolve a DNS name.

Which resolvers are capable of posing queries in IPv6? Figure 10 shows the top 25 IPv6-capable visible resolvers, grouped according to the origin network (Origin AS) where they are announced.

AS15169	31.9%	GOOGLE - Google Inc., USA
AS7018	13.5%	ATT-INTERNET4 - AT&T Services, Inc., USA
AS7922	11.5%	Comcast Cable Communications, LLC, USA
AS36692	3.4%	OPENDNS - OpenDNS, LLC, USA
AS8151	2.7%	Uninet S.A. de C.V., MX Mexico
AS17676	2.4%	GIGAINFRA Softbank BB Corp., Japan
AS4134	1.7%	CHINANET-BACKBONE No.31,Jin-rong Street, China
AS28573	1.6%	CLARO S.A., Brazil
AS9498	1.6%	BBIL-AP BHARTI Airtel Ltd., India
AS3320	1.4%	DTAG Internet service provider operations, Germany
AS2516	1.2%	KDDI KDDI CORPORATION, Japan
AS6147	1.1%	Telefonica del Peru S.A.A., Peru
AS18881	1.0%	TELEFONICA BRASIL S.A, Brazil
AS22773	1.0%	Cox Communications Inc., USA
AS55836	1.0%	RELIANCEJIO-IN Reliance Jio Infocomm Limited, India
AS55644	0.9%	IDEANET1-IN Idea Cellular Limited, India
AS6713	0.9%	IAM-AS, Morocco
AS4713	0.9%	OCN NTT Communications Corporation, Japan
AS6128	0.9%	CABLE-NET-1 - Cablevision Systems Corp., USA
AS20115	0.8%	CHARTER-NET-HKY-NC - Charter Communications, USA
AS3352	0.8%	TELEFONICA_DE_ESPANA , Spain
AS852	0.8%	ASN852 - TELUS Communications Inc., Canada
AS22394	0.5%	CELLCO - Cellco Partnership DBA Verizon Wireless, USA
AS6799	0.5%	OTENET-GR Athens - Greece, Greece
AS15557	0.4%	LDCOMNET ,France

Figure 10 – TOP 25 IPv6-capable visible resolvers, grouped by Origin AS

The widespread use of Google's Public DNS service is clearly evident here, and one half of the DNS v6 capability is due to the use of the three resolver sets provided by Google, AT&T and Comcast.
If some 35% of users send their queries to resolvers who are capable of using IPv6 to ask authoritative servers, what is the proportion of IPv6 queries that one can expect at a dual stack authoritative name server?

We can use the query profile from the "parent" authoritative name server to answer this. Over the course of the experiment we saw some 3,113M queries to this name server, and of these some 325M were made using IPv6. In other words, in terms of query profile, the dual stack name server saw some 11% of its queries being made using IPv6.

If dual stack resolvers were to make a random selection of which protocol to use we might expect this number to be higher, and we might expect to see some 17% of queries using IPv6. The difference could be due to the local environment, where users may order their local resolver set to preference a local IPv4-only resolver and fall back to a dual stack resolver, such as Google's Public DNS service in the event of response timeout.

It appears from these results that there is a lot of IPv6 in the DNS. Perhaps more than we anticipated. But that does not necessarily mean that you should turn on IPv6 in your DNS infrastructure without considerable care and attention to detail. We still see a significant level of IPv6 Path MTU black holes, so it makes some sense to clamp the TCP MSS down to 1220 on IPv6 TCP servers to try to avoid this problem in the first place. At the same time, we also see a significant level of IPv6 extension header packet drop, and in a previous experiment we observed a 30% drop rate for fragmented IPv6 DNS responses. So that implies that for UDP it makes some sense to keep the UDP MTU high. In the case of this experiment we used a 1,500 octet UDP MTU and this probably is a major factor why we saw negligible levels of packet drop for a 1,425 octet response. The usual caveats about avoiding 6to4 and other forms of 6-in-4 tunnelling apply here. You should try to use native mode IPv6 for infrastructure services such as the DNS in order to avoid the unwanted and unintended side effects of tunnelling.

IPv6 in the DNS

The conclusion appears to be that when we compare the use of IPv6 in the world of web objects to that of the infrastructure of the DNS, the DNS has seen significant progress in the adoption of IPv6, and slightly more than one third of all users in today's Internet are capable of resolving names using IPv6, as compared with a 7% measurement of users capable of using IPv6 in fetching objects over the web. The DNS is well on the path of transition and perhaps further along this path than all the other elements of the Internet's infrastructure.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region. He has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for building the Internet within the Australian academic and research sector in the early 1990's. He is author of a number of Internet-related books, and was a member of the Internet Architecture Board from 1999 until 2005, and served on the Board of Trustees of the Internet Society from 1992 until 2001 and chaired a number of IETF Working Groups. He has worked as an Internet researcher, as an ISP systems architect and a network operator at various times.

www.potaroo.net

Disclaimer