The ISP Column
A monthly column on things Internet

November 2012

Geoff Huston

Counting IPv6 in the DNS

At the recent ARIN XXX meeting in October 2012 I listened to a debate on a policy proposal concerning the reservation of a pool of IPv4 addresses to address *critical infrastructure*. This term is intended to cover a variety of applications, including use by public Internet Exchanges and authoritative nameservers for various top level domains. As far as I can tell, the assumptions behind this policy proposal includes the assumption that a top level authoritative nameserver will need to use IPv4 for the foreseeable future, so that an explicit reserved pool of these IPv4 addresses needs to be maintained for use by the authoritative nameservers for these domain names. But it this really the case? If you set up an authoritative DNS nameserver for a domain name where all the nameservers were only reachable using IPv6, then what is the visibility of this nameserver? What proportion of the Internet's user base could still access the name if access to the authoritative nameservers was restricted to only IPv6?

There are three questions that would be useful to answer in this context:

1. What proportion of DNS resolvers are capable of performing DNS queries using IPv6?

2. What proportion of users are using IPv6-capable DNS resolvers?

3. Can we see evidence of IPv6 packet fragmentation handling issues when we construct large responses with DNSSEC?

The Experiment

This is another experiment relating to testing the capabilities of end users on the Internet, and it is once more an ideal experiment for enlisting the assistance of an online ad delivery network.

In this case what we have designed is an advertisement with an associated block of Flash code. When the ad is delivered into the user's browser the ad's code will generate a pair of unique DNS names, which will be then used in a pair of URLs which will then be fetched by the client. The pair of URLs used in this experiment is:

```
http://t10000.u7579899479.s1348442285.i767.v6022.e5fa44f2b31c1fb553b6021e7360d07d5d91ff5e.f.t7.
     dotnxdomain.net/1x1.png

http://t10000.u7579899479.s1348442285.i767.v6022.5fa44f2b31c1f.g.t7.dotnxdomain.net/1x1.png
```

The significant parts of this URL are :

`u7579899479.s1348442285` - a unique identifier string that binds the web object fetches, the DNS name queries performed in a single experiment together. This string is unique in order to ensure that various forms of caching of both DNS responses and Web objects is not performed by network middleware. Every ad impression essentially generates a new DNS query question whose answer has not been pre-loaded into any DNS cache, and a new web fetch where the named object does not reside in any web object cache. The `u` field is a semi-random string, while the `s` field is the time as measured by the number of seconds since 1 January 1970 UTC.

`v6022` - the experiment version

`t7` - this is a domain whose authoritative nameserver is accessible only via IPv6

The terminal objects themselves (the 1x1.png web objects) are accessible using IPv4 only. Indeed all parts of this experiment are deliberately positioned as IPv4-only network elements with one exception: the `t7.dotnxdomain.net` zone uses nameservers that are IPv6 only.

The DNS configuration for this experiment is as follows. The `dotnxdomain.net` is a DNSSEC signed domain. The domain has two nameservers, both which resolve to IPv4 addresses that are address aliases on a single authoritative nameserver. The subdomain `t7.dotnxdomain.net` is a DNSSEC signed subdomain of `dotnxdomain.net`. The domain is served again by two nameservers, but in this case these names resolve to IPv6-only addresses that are again aliases on a single authoritative name server. This name server is hosted on a distinct server platform. The `t7.dotnxdomain.net` uses a wildcard entry with an IPv4 A resource record.

The other part of this setup is the effort to investigate the behaviour of the DNS query process when the IPv6 response to the IPv6 query directed to the `t7.dotnxdomain.net` nameserver will not fit into a single IPv6 UDP response packet. These zones are DNSSEC-signed. The experiment was structured such that the intended IPv6 responding packet size for the `f.t7` query was 1520 octets, so that the response was intended to fit in a single IPv6 packet, while the `g.t7` response was structured to be 1480 octets, so that IPv6 packet fragmentation is required from the outset.

The experiment used two server systems, running FreeBSD 8.1. One system was configured with Bind 9.9.1-P2 and two IPv6 addresses. This system was configured to perform both DNS query logging and IPv6 packet capture logging. This system was configure as the authoritative name server for `t7.dotnxdomain.net`. The second systems also used FreeBSD8.1 and Bind 9.9.1-P2, as well as the Apache 2.2.17 http server. This system was configured with http, DNS and packet capture logging.

The experiment was active from 21 September 2012 to 27 September 2012.

Resolvers that can use IPv6

The first is the question relating to DNS resolvers and their capability to undertake DNS queries using the IPv6 protocol.

How many DNS resolvers generated queries in this experiment over IPv4?
111,538

How many DNS resolvers also generated queries in this experiment over IPv6?
5,225

That's **4.7%** of the set of visible DNS resolvers who are showing that they are capable of performing queries using IPv6.

For comparison, I see that some 1.6% of visible DNS resolvers appear to be DNSSEC-validating resolvers (http://bit.ly/U1OfTF), so this figure of 4.6% is not a bad outcome, and shows an elevated awareness of the need to support dual stack query resolution in the DNS.

In looking at this count of resolvers that perform DNS queries over IPv6, the count used here is a count of unique IPv6 addresses. However, in IPv6 it is possible for a system to turn on "privacy addresses", which enables a host to change the low order 64 bits of its presented interface identifier address from time to time. There is no clear bit signature of a privacy addresses, so there is a certain amount of guided guess work to estimate the number of actual systems that lie behind these 5,225 IPv6 addresses.

IPv6 privacy addresses are not readily identified as there is no bit field in the interface identifier part of the address that denotes the interface identifier as a privacy-generated identifier. The procedure I used to identify IPv6 privacy addresses from the set of DNS resolver addresses included identifying those IPv6 addresses that:
- do not use `0xfffe` in bits 24 though 40 in the interface identifier,
- have set bit 6 to `0` in the interface identifier,
- have non-zero nibble values in each of the 4 nibbles in the interface identifier, and
- where there are two or more such addresses with a common 64 bit network identifier.

Of these 5,225 addresses some 9 addresses appear to be privacy addresses used by 2 distinct host systems. In other words, it appears that we actually see 5,218 distinct resolvers that are capable of performing DNS queries using IPv6, which still represents some 4.7% of the set of all seen resolvers.

We can use a form of geo-location to map these resolver addresses into countries. We can also weight each resolver by the number of clients who used this resolver. Because this experiment uses both IPv4 and IPv6 we can derive the relative proportion of DNS resolvers that are capable of using IPv6, as shown below in Table 1. The full list of the countries where visible resolvers were found, and their weighted proportion of IPv6-capable DNS resolvers can be found at: http://bit.ly/R6RXdI .

CC	%v6	V6 Clients	V4 Clients	Country
BT	124%	158	127	Bhutan (*)
JE	95%	57	60	Jersey
LI	79%	43	54	Liechtenstein
HU	66%	16,717	24,969	Hungary
EE	56%	1,343	2,380	Estonia
SI	56%	3,819	6,771	Slovenia
LV	54%	1,687	3,120	Latvia
TH	49%	100,694	201,883	Thailand
FO	47%	19	40	Faroe Islands
CZ	45%	4,429	9,740	Czech Republic
PT	42%	8,776	20,576	Portugal
DE	40%	14,202	34,950	Germany
US	40%	465,169	1,145,319	United States of America (**)
ZM	39%	265	676	Zambia
UG	36%	1,353	3,749	Uganda
LU	33%	909	2,705	Luxembourg
SE	31%	3,614	11,368	Sweden
HR	30%	7,878	25,490	Croatia
ID	28%	16,219	56,762	Indonesia
JP	27%	55,314	198,785	Japan

*   Some of the V4 resolvers are announced from an AS registered to a different CC code
** AS15169 (Google's global Public DNS service) is included in the US figures

Table 1: Weighted Proportion of IPv6-capable DNS resolvers, by Country – Top 20 Countries

This ranking is perhaps a little misleading is so far as that there are a number of countries with quite small counts of visible resolvers (and, unless you are looking at something like a national happiness index, may well be one of a very small number of national rankings that has Bhutan at the top of the list!), which means that a small number of resolvers in a given country may produce quite high IPv6 outcomes. The Faroe Islands, Jersey and Lichtenstein are examples of countries with relatively small numbers of sample points.

Another way of looking at this resolver data is to look at the count of clients using these DNS resolvers by the Origin AS of the resolver. Those resolvers that were used the most in the context of this experiment, ordered by the Origin AS of the resolver are listed in Table 2 below. The complete list of origin AS's that had visible resolvers can be found at http://bit.ly/PV1s4W

Weighted IPv6	Weighted Ipv4	Origin AS	AS Name
383,742	324,968	AS15169	GOOGLE - Google Inc., USA
63,344	51,998	AS45758	TRIPLETNET-AS-AP TripleT Internet, Thailand
38,954	91,186	AS7922	COMCAST-7922 - Comcast Cable Communications, Inc., USA
34,072	58,877	AS9737	TOTNET-TH-AS-AP TOT Public Company Limited, Thailand
21,453	51,389	AS4713	OCN NTT Communications Corporation, Japan
16,308	14,337	AS8708	RDSNET RCS & RDS S.A., Romania
15,746	12,609	AS2518	BIGLOBE NEC BIGLOBE, Ltd., Japan
15,415	20,048	AS12322	PROXAD Free SAS, France
13,824	13,062	AS5483	HTC-AS Magyar Telekom plc., Hungary
11,850	27,322	AS17974	PT Telekomunikasi Indonesia, Indonesia
9,736	12,105	AS3320	DTAG Deutsche Telekom AG, Germany
9,351	36,386	AS36692	OPENDNS - OpenDNS, LLC, USA
7,629	8,576	AS22773	ASN-CXA-ALL-CCI-22773-RDC - Cox Communications Inc., USA
7,443	5,412	AS7018	ATT-INTERNET4 - AT&T Services, Inc., USA
7,435	8,527	AS3243	TELEPAC PT Comunicacoes, S.A.,Portugal
6,054	962	AS6939	HURRICANE - Hurricane Electric, Inc., USA
5,826	14,064	AS5391	T-HT Hrvatski Telekom d.d., Croatia
4,922	6,273	AS6327	SHAW - Shaw Communications Inc., Canada
4,584	4,610	AS10030	CELCOMNET-AP Celcom Internet Service Provider, Malaysia
4,549	5,810	AS9824	ASN-ATHOMEJP Technology Networks Inc., Japan

Table 2: Weighted Proportion of IPv6-capable DNS resolvers, by Origin AS – Top 20 AS's

Clients that use Resolvers that can use IPv6

Now lets turn our attention to the second question, namely counting the number of clients who use DNS resolvers that are capable of resolving DNS names using IPv6.

There are two ways of looking at the client counts. Firstly, in this experiment each client needs to resolve the experiment's DNS strings into IP addresses, and we can count the number of clients that successfully pose the DNS query using IPv6. Secondly, the client uses the returned addresses to perform web fetches of the named object, and we can count the number of clients who successfully fetch the web object. (It should be noted that the web objects here are all IPv4-based web objects. It is only the DNS name itself that uses IPv6-only authoritative nameservers). As the experiment is operated under an impression of an online advertisement we find that a significant proportion of experiments do not run through to completion, so it may be worth splitting out the per-client measurements that use data extracted from the DNS query logs and data extracted from the web server logs.

In terms of the DNS data we have the following results:

How many client experiments completed DNS queries?

2,300,384

How many client experiments completed IPv6 DNS queries?

432,632 or **19%**

The attrition rate from DNS query phase to performing web fetches is quite high, as the web fetch logs show the following:

How many unique IP addresses completed web fetches for objects named in the experiment?

890,920

How many clients were able to perform web fetches that required IPv6 DNS resolvers?

161,125 or **18%**

As with the resolver data, we can break out the relative proportion of these clients who can perform DNS queries over IPv6 and generate a coloured map of the world to illustrate this data, as shown in Figure 1. The data used to generate this map can be found at http://bit.ly/Tl0HkN.

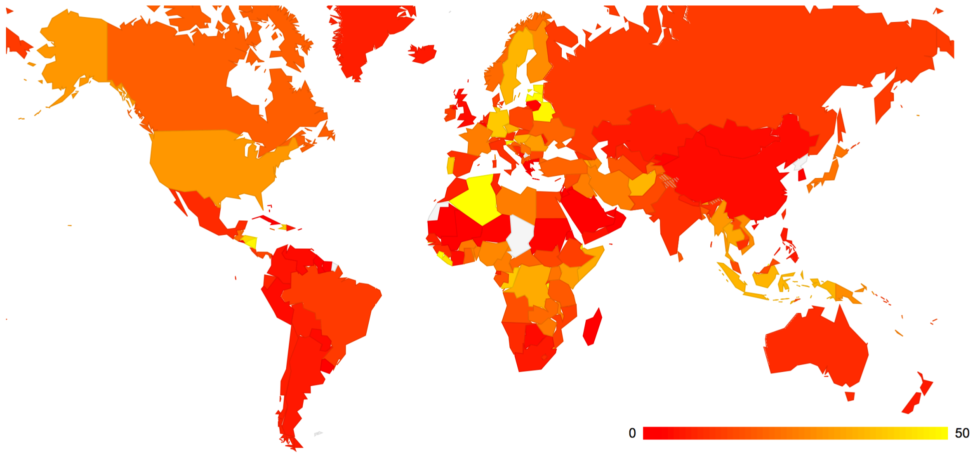

Figure 1: Weighted Proportion of clients using IPv6-capable DNS resolvers, by Country

While the experiment was executed some 2.3 million times the distribution of experiments across many countries means that some countries have relatively low levels of representation. The table below takes this data and uses a threshold of a minimum of 500 sample points within each country, which encompasses 111 countries, to select a subset of countries, which are then ranked in the following table. The top 26 countries, ranked by the proportion of clients who use IPv6-capable DNS resolvers is as follows:

% of Clients with IPv6-capable DNS resolvers	Count of clients with IPv6-capable Resolvers	Clients	Country
52.08%	676	1,298	Occupied Palestinian Territory
50.44%	1,710	3,390	Algeria
49.54%	590	1,191	Latvia
48.90%	1,540	3,149	Belarus
48.88%	1,048	2,144	Slovenia
47.29%	514	1,087	Estonia
39.36%	3,520	8,943	Germany
39.14%	2,591	6,619	Portugal
36.15%	1,486	4,111	Singapore
36.12%	7,769	21,509	Indonesia
35.70%	623	1,745	Sweden
35.05%	184	525	Luxembourg
34.52%	1,240	3,592	Czech Republic
34.38%	3,342	9,721	Hungary
32.89%	11,232	34,152	Thailand
31.34%	874	2,789	Armenia
31.08%	5,748	18,497	Romania
31.07%	933	3,003	Kenya
30.06%	11,006	36,616	USA
27.58%	1,710	6,201	Vietnam
27.46%	299	1,089	Finland
26.90%	202	751	Nigeria
26.87%	632	2,352	Azerbaijan
25.07%	285	1,137	Iraq
25.02%	3,697	14,778	France

Table 3: Proportion of Clients using IPv6-capable DNS resolvers, by Country – Top 26 countries

Again, we can also look at this data set using the origin as of the client's IP address, or the client's network provider instead of the country code. This data can be found at http://bit.ly/R7KmvD. Using a threshold of 50 or more sample points per origin AS, in order to remove the under-sampled AS's, we are left with the following ranking of networks in terms of the proportion of clients who use a set of DNS resolvers that includes the capability to resolve DNS queries using IPv6.

%	AS	V6-DNS Clients	Clients	AS Name
89%	AS52242	50	56	Yota De Nicaragua, Nicaragua
89%	AS15169	147	165	GOOGLE - Google Inc., United States of America
88%	AS28545	52	59	Cablemas Telecomunicaciones SA de CV, Mexico
88%	AS28220	78	89	, Brazil
87%	AS28509	95	109	Cablemas Telecomunicaciones SA de CV, Mexico
86%	AS38844	51	59	NTNU-TW National Taiwan Normal University, Taiwan
86%	AS28516	72	84	Cablemas Telecomunicaciones SA de CV, Mexico
85%	AS36991	53	62	ORANGE-UG, Uganda
85%	AS42248	52	61	VIDA-OPTICS Vida Optics TVV, Bulgaria
85%	AS28512	46	54	Cablemas Telecomunicaciones SA de CV, Mexico
85%	AS53006	252	296	, Brazil
85%	AS262227	106	125	Claro Panam· S.A., Panama
84%	AS21804	54	64	Access Communications Co-operative Limited, Canada
84%	AS39309	54	64	EDUTEL-AS Edutel B.V., Netherlands
83%	AS11814	278	333	DISTRIBUTEL COMMUNICATIONS LTD., Canada
83%	AS7922	5,743	6,902	Comcast Cable, United States of America
83%	AS3243	2,385	2,872	TELEPAC PT Comunicacoes, S.A., Portugal
83%	AS52075	62	75	WIFIRST Wifirst S.A.S., France
82%	AS15975	497	609	Hadara Technologies, Occupied Palestinian Territory
82%	AS198471	71	87	LINKEM-AS Linkem spa, Italy
82%	AS35063	62	76	TKCHOPIN-AS TKChopin Computer Centre, Poland
81%	AS5645	365	448	TEKSAVVY-TOR TekSavvy Solutions Inc. Toronto, Canada
81%	AS25441	82	101	IBIS-AS Imagine Group Ltd., Ireland
81%	AS29084	182	225	COMNET-AS Comnet Bulgaria Holding Ltd., Bulgaria
80%	AS49363	275	343	OAR-DC "Orange Armenia" CJSC, Armenia
80%	AS42689	56	70	Cablecom Networking Limited, United Kingdom

Table 4: Proportion of Clients using IPv6-capable DNS resolvers, by Origin AS – Top 26 AS's

IPv6 Path MTU Problems?

Can we see evidence of IPv6 packet fragmentation handling issues when we construct large responses with DNSSEC?

One of the major changes between the IPv4 to IPv6 protocols is the change in the treatment of packets that are too large to forward along a network link. In IPv4 the function is for the router to fragment the large IPv4 packet into units that will fit into the link, and then forward all the fragmented parts toward the destination. IPv6 does not attempt to undertaken fragmentation on the fly, and instead the router in question is required to generate an ICMP6 packet back to the packet's sender, including the diagnostic code of "packet too big", the packet size that would allow the packet to be passed into the link, and the header bytes of the original packet header.

When the sender receives this ICMP message it should note this new maximum message size for this destination, and, as appropriate, it may chose to resend the original data using packets no larger than this revised message size, and thereafter send further packets to this destination within the parameters of this reduced message size.

For the TCP transport protocol this could be considered to be reasonable behaviour. For the UDP protocol this is a problem. At the transport protocol level the UDP sender has no packet memory. The transaction is complete when the packet is sent, so the ICMP response cannot elicit a retransmission from the UDP transport protocol. For UDP applications, such as the DNS, this can present certain performance problems. The packet fragmentation event does not generate retransmission of the original packet, so the client will be forced to time out and attempt further queries.

We constructed this experiment with the assumption that the nameserver software we are using would be generating 1500 octet packets in UDP, and we thought we might find instances of resolution failure due to this issue of fragmentation handling in IPv6. We were also looking for instances of ICMP6 packet filtering, which would cause resolver-side timeouts and repeated queries.

However, in this case our assumptions about the behavior of the name server we were using were incorrect. The issues of supporting UDP transactions in IPv6 is a well known problem, as written up in the Internet-draft *draft-andrews-dnsext-udp-fragmentation*. The corrective measure used in the Bind 9.9.1 authoritative nameserver software in the FreeBSD platform I used in this experiment is not to send any UDP packets larger than the IPv6 minimum MTU of 1280 octets. Accordingly, we saw no instances of ICMP packet too big messages in response to UDP answers.

Does that mean that the issue of IPv6 packet fragmentation and the problems raised by ICMP6 filtering has been eliminated in DNS over IPv6?

Not exactly.

DNS can operate over both UDP and TCP. The fallback to TCP occurs when a response is truncated in UDP, and the resolver may then use TCP to obtain the complete response. In this case all queries that include the EDNS0 DNSSEC capable flag will cause the Bind name server to generate responses greater than 1280 octets, so all of these responses will be fragmented in UDP. However, it has been noted that many firewalls block trailing fragments, so that the UDP response will generate an incomplete response, which may trigger a re-query in TCP. In this experiment, out of the 432,632 experiments that successfully queried the authoritative nameserver using UDP over IPv6, some 45,760, or some 10.5% of queries, also repeated the same query using TCP.

Again, because we have deliberately made the DNSSEC DNS response for at least on of the queried names to be greater than 1500 octets then we expect the TCP session to send at lease one packet of the maximal TCP session size. The nameserver was deliberately set up using a local interface MTU of 1500, so that when the TCP session opened up it offered a MSS of 1440 to the remote DNS resolver. There is the possibility that the IPv6 path between the local authoritative DNS nameserver and the remote DNS resolver contains link elements with lower MTU sizes (such as found in various forms of tunnel encapsulation), so that we expect to see a certain level of path MTU adjustment from these 45,760 TCP sessions.

And this happened for some DNS over TCP responses. Within the scope of this experiment we received 4,670 ICMP packet too big ICMP messages in response to the large TCP response packet. This implies that in 10% of the cases in the end-to-end IPv6 path from the authoritative name server to the DNS resolver there was a path element that was unable to accept these 1500 octet-sized packets.

As noted above, the ICMP response includes a 32 bit field that nominates the actual MTU of the link that caused the ICMP packet too big message to be generated. We saw the following set of MTU sizes in the set of received ICMP messages:

Message count	Received MTU
4	1280
19	1476
265	1480
4,382	1500

The 4 messages with the 1280 response appear to be from tunnels that use the IPv6 minimum MTU. The 19 messages with a 1476 MTU appears to be using a form of tunnelling over IPv4 using a 24 octet IPv4 packet header, which is 20 octets of IPv4 packet header and a 4 octet option or padding word. The 265 messages with a 1480 MTU appear to be using a conventional protocol 41 IPv6 in IPv4 tunnel, with the 20 bytes being used for an IPv4 packet header.

However the 4,382 messages that respond with a 1500 MTU are simply broken! These ICMP messages are saying, in effect, that the 1500 octet packet it attempting to forward were is too big for the next link, and the sender should try instead to use a 1500 octet packet. Clearly this is not going to work.

Where are these routers that are generating these broken forms of ICMP Path MTU control messages? The following table lists these broken IPv6 routers and the origin AS.

#msgs	router	CC	AS, AS NAME
62	2001:620:610:20::20	CH	AS559, Swiss Education and Research Network
12	2001:630:0:9003::2	GB	AS786, JANET The JNT Association
4	2001:630:53:89c4::26	GB	AS786, JANET The JNT Association
8	2001:660:3305:a205::111	FR	AS2200, RENATER
2	2001:6a8:2500:1000::2	BE	AS2611, BELNET
73	2001:c18:0:3001::4	MY	AS10204, ARCNET-NTT
102	2001:c38:9004:6::2	BE	AS2611, Communication Authority of Thailand
3649	2001:c68:bfff:5::d	CN	AS4134, CHINANET-BACKBONE
69	2001:ff8:1:254::24	MO	AS7582, University of Macau
26	2001:1284:ff00:ffff::4	BR	AS14868, Companhia Paranaense de Energia - COPEL
10	2001:14f0:0:5::e	DE	AS12355, HHeLi NET Telekommunikation GmbH & Co.
10	2001:49b8::a	US	AS21737, SPRINGNET2-NET - SpringNet
55	2401:b000:2::a	MY	AS17971, TMVADS-AP TM-VADS DC Hosting
294	2605:f000::3	US	AS22442, PHONOSCOPE
6	2a00:dc8:0:f::4	NL	AS39637, Netlogics BV

Table 5: Routers responding with 1500 octet ICMP6 Packet too big messages

Is this a critical for clients using DNS resolvers that sit behind these routers? Do the clients who are attempting to resolve these DNS names and encounter this particular IPv6 path MTU problem manage to fail over to other DNS resolvers and complete the DNS resolution?

The answer is a mixed one. We observed 3,077 separate experiments where the TCP response generated these anomalous ICMP6 packet too big messages. In approximately half of the cases, 1,445 experiments, the client appears to fail over to another V6 capable DNS resolver and complete the DNS resolution and the subsequent web fetch. In the other 1,632 cases the web fetch does not take place, which in many cases may be attributable to DNS resolution failure due to this anomalous treatment of the ICMP6 packet too big message by these routers.

Conclusion

This is in many ways an experiment that confirms our current understanding of the state of play with deployment of IPv6. The service providers managing the Internet's transit transmission infrastructure, and much of the Internet's service infrastructure, including name resolution, is well abreast of the need to convert into dual stack support, and we are seeing relatively high ratios of IPv6 capability in these aspects of the network.

The experiment's results indicate that as of September 2012 some 18% of today's clients appear to use DNS resolver configurations that include individual resolvers that are capable of undertaking DNS queries using IPv6 to perform the DNS query. This result is consistent with the general observation about IPv6 deployment proceeding apace in the infrastructure components of the Internet.

But if this result paints a positive story about the state of IPv6 deployment in the infrastructural elements of the internet, the same cannot be said about the Internet's last mile access networks. We have been using a similar technique to measure the relative number of end host systems that will prefer to use IPv6 in a dual stack scenario, and the Internet-wide metric of visible end systems who exhibit IPv6 capability is far lower. At the same time as we see an 18% figure of IPv6 capability in the DNS, only some 0.18% of today's end client systems will use IPv6 to actually fetch a dual stack object (http://labs.apnic.net/dists/v6dcc.html).

Afterword

Nothing is ever what it seems to be! Those strange ICMP6 packet too big messages that I thought were in response to a large TCP packet - well they weren't being generated by the TCP packet.

I had thought that if my authoritative nameserver was sending out packets no larger than 1280 octets in UDP then any incoming ICMP6 Packet Too Big message could not be received from a UDP packet whose size was no larger than the minimum MTU defined in IPv6. After all IPv6 routers need to be able to carry without fragmentation a packet whose size is 1280 octets.

But I was wrong. Alexander Gall kindly pointed out to me that if a certain vendor's firewall product was lagging from the current version of the vendor's software, then the firewall could indeed perform this feat and generate these strange ICMP6 packet too big messages in response to small-sized UDP packets! There is a sort-of-logical explanation of why the firewall is nominating a 1500 octet MTU as well.

The firewall software evidently has a DNS ALG "feature". When this feature was turned on the firewall performed a reassembly of the UDP fragments of the DNS response. It then applied the firewall filter rules against the reassembled packet to see if the DNS response was acceptable. Now if the firewall decided that the DNS response was to be allowed through, it appears that it did not pass through the original UDP fragments of the DNS response, but attempted to pass through the reassembled UDP over IPV6 packet. At this point the forwarding engine could not cope with this artificially large packet, and it performed a conventional IPv6 handling of a large packet: generating an ICMP6 packet too big message back to the sender and discarding the packet. And because the link uses a 1500 MTU, then it dutifully placed the value 1500 into the ICMP6 message.

Now the resolver behind this oh-so-helpful firewall now sees no response whatsoever, and after a number of repeated queries it might time out and try TCP. But this time when the TCP response is sent back it generates no ICMP6 packet too big messages. In this case the authoritative nameserver sends the two TCP packets that carry the DNS response back to back, where the first packet is a full-size 1500 octet packet. But this large DNS response does not generate an ICMP6 packet too big response. Apparently the firewall has managed to do the correct thing in this case and forward on the TCP packets as received if it accepts the DNS response.

So if you are the lucky owner of one of these firewalls, then perhaps you should look at the vendor's bug list and software versions and check if the version of SRX firewall software you are running has had this bug in the IPv6 part of the DNS ALG "feature" corrected.

Disclaimer

About the Author

Geoff Huston B.Sc., M.Sc., has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and has been active in the Internet Engineering Task Force for many years.

www.potaroo.net