The ISP Column
A column on various things Internet

March 2024
Geoff Huston

Adding IPv6-only to DNS and Truncation in UDP

In February I looked at the behaviour of the DNS when processing responses in UDP which set the Truncated flag in the DNS response. In particular, I was looking for the incidence of DNS resolvers which used the Answer section in truncated responses (despite the admonition in DNS standards not to do so) and the extent to which there are DNS resolvers out there that are incapable of using DNS over TCP.

This month I'll report on a repeat of this experiment using a test environment where only IPv6 can be used.

To briefly recap, the DNS leverages the UDP transport protocol to maximise its efficiency. UDP transport allows servers to support a far higher UDP query load as compared to TCP queries and responses. The issue with UDP is that while the underlying IP specification may permit IP packets of up to 65,535 octets in size, most networks operate with a far lower maximum packet size. When an IP device attempts to forward a packet onto a network that is too large for the network then it will need to *fragment* the IP packet. In IPv4 this fragmentation can be handled *on-the-fly*, and the resulting fragments are reassembled at the destination. IN the case of IPv6 the packet causes an ICMPv6 Packet Too Big control message to be passed back to the packet's source address, and it is left to the sender to fragment outgoing packets to this destination that are fragmented at the source.

If IP networks were totally reliable in delivering fragmented packets, then applications such as the DNS, could potentially leverage this reliability and send all messages up to this 65,535-octet limit over UDP. Larger messages, such as zone transfers using AXFR would need to use UDP, but all other messages could use UDP, fragmenting the message as required. However, there is another limitation associated with large IP packets, and that is the amount of memory that a host is prepared to use in reassembling fragmented packets. Packets which are larger than this host limit are discarded. In the specification of IPv4 the minimum maximum reassembly buffer size for hosts is 576 octets.

Accordingly, the initial design of the DNS steered a conservative path through this space, and no DNS payload using UDP was permitted to be larger than 512 octets. If a DNS responder wanted to pass a larger message it would send a *truncated* UDP response that was deliberately chopped such that the response was no larger than 512 octets and set the Truncated bit in the response (conveniently located in the Header section of the DNS response). A DNS client, upon receiving a UDP response with the truncated bit set was expected to re-query using TCP.

In 2013 RFC 6891, "Extension Mechanisms for DNS" was published. This mechanism allowed the querier to specify that it was capable of reassembling IP fragments of packets larger than 512 bytes in the query, allowing the responder to send responses up to this size in UDP. To quote this RFC "A good compromise may be the use of an EDNS maximum payload size of 4096 octets". As far as I can tell, the thinking behind this advice was that using fragmented UDP was reasonably reliable, and the overheads of re-querying over TCP were considered to impose a higher cost relative to the risks of loss of fragmented UDP packets.

Unfortunately, many networks are really not very good at all in delivering fragmented IP packets. In IPv4 the problem appears to be the use of security firewalls that find that IP packets that contain trailing fragments represent an unacceptable security risk. In IPv6 the problem is further compounded with the use of an additional fragmentation extension header, which appears to cause some active network elements to drop the packet (https://stats.labs.apnic.net/v6frag).

The current DNS approach is to avoid packet fragmentation and do so by setting the EDNS buffer size of 1232 octets (https://www.dnsflagday.net/2020/). When a DNS response is larger than this size, then it will need to truncate the UDP response, triggering the DNS querier to re-query over TCP.

RFC 2818 has some clarifying advice on what a standards-compliant implementation should do when it receives a truncated response:

> "When a DNS client receives a reply with TC set, it should ignore that response, and query again, using a mechanism, such as a TCP connection, that will permit larger replies."
> RFC 2191, Clarifications to the DNS Specification, July 1997.

This RFC does not provide any reasons why the clarification was considered to be necessary. It is likely that there was a view at the time that some recursive resolver implementations were trying to improve their responsiveness by acting in an opportunistic manner with truncated DNS responses. Even if the truncated bit flag was set, if there was a usable answer section in the response then the resolver might use the provided data and move on with the resolution task if possible. This optimisation applies particularly to DNS referral responses where the answer section is complete and intact and the glue records in the additional sections in the response may be truncated.

This leads to two questions that relate to the current DNS resolution environment:

- How prevalent is the behaviour of using the data in a truncated DNS response?
- Do all name resolution systems successfully followup with a re-query using TCP after receiving a truncated UDP DNS responses?

The earlier report looked at answers to this question using a dual-stack environment. In this report I would like to report on some further measurements using an IPv6-only DNS server environment.

How much of the DNS supports IPv6-only Servers?

As usual asking quantification questions of the DNS is not straightforward. Are we talking about DNS authoritative servers? Or DNS recursive resolvers? Or DNS queries? Or DNS zones? In this work we use a basic metric of end users, using an ad-based measurement framework to sample a large volume of end user capabilities every day.

So, in this case we are looking at the proportion of users who are located behind DNS resolution infrastructure that can handle on IPv6-only authoritative servers. Not all of the DNS resolution infrastructure is dual stacked, as shown in Table 1.

	Dual Stack	V6-Only
Tests	67,469,670	134,295,458
Responses	67,469,670	92,606,626
Ratio	100.00%	68.96%

Table1 – IPv6-Only Capability

Some 30% of users are behind IPv4-only DNS resolution infrastructure, and cannot resolve a DNS name if the servers are configured as IPv6-only servers.

How do DNS Resolvers behave with Truncation?

We conducted this test February 2024 and March 2024. The results are shown in Table 2.

	Dual Stack	V6-Only
Tests	67,469,670	92,606,626
xTC	33,026,054	46,303,113
xTC-noTCP	306,271	1,311,313
Query Target	78,777	6,718
Rate	0.239%	0.015%

Table 2 – Test of DNS Truncation

In the test framework one half of the users were handed back a response with an empty answer section with the truncation bit set, while the other half was handed back a normal response with an Answer section and an Additional section with glue records which was under 512 octets, with the truncation bit set. This is the **xTC** test in Table 2. For Dual Stack tests 99% of cases perform a follow-up query over TCP, while in IPv6-only the rate is slightly smaller at 97%. What we are looking for here is to find cases where the resolver uses the information contained in the truncated response to learn the IP address of the name server for the query target and then query for the target name. In IPv4 the incidence of this situation is in 0.239% of cases, while in IPv6 the ratio is significantly lower, at 0.015% of cases.

An uninformed guess as to the reason why the IPv6 number is lower than the dual-stack rate is that IPv6-capable resolvers are more recent in terms of the time of installation as compared to some of the old, but still in-service IPv4 resolvers.

However, the bottom line is a positive outcome for IPv6 here, where most IPv6-capable resolvers do not use the information contained in a truncated response.

Rank	AS	CC	Count	Rate	AS Name
1	17882	MN	5,021	60.06%	UniVision, Mongolia
2	17816	CN	957	11.45%	China Unicom, Guandong, China
3	4837	CN	557	6.66%	China Unicom, Backbone, China
4	36923	NG	495	5.92%	SWIFTNG, Nigeria
5	4134	CN	444	5.31%	ChinaNet, China
6	4538	CN	185	2.21%	CERNET, China
7	4812	CN	93	1.11%	Chinanet, China

Table 3 – Networks that show use of IPv6 Truncated Responses

As shown in Table 3, some 5,000 cases (60% of all such cases) where the resolver appears to be using truncated DNS response data occur for users located in UniVision, an ISP located in Mongolia.

Finally, is this an issue for all IPv6-capable resolvers within these three networks, or a more isolated set of behaviours? Let's add a couple of columns to Table 3, looking at the total number of samples drawn from each of these three networks, and the proportion of these samples that were observed to make use of the data contained in truncated responses. This is shown in Table 4.

Rank	AS	CC	TC Count	Sample Count	Rate	AS Name
1	17882	MN	5,021	36,802	13.64%	UniVision, Mongolia
2	17816	CN	957	108,012	0.89%	China Unicom, Guandong, China
3	4837	CN	557	3,035,580	0.02%	China Unicom, Backbone, China
4	36923	NG	495	36,923	1.34%	SWIFTNG, Nigeria
5	4134	CN	444	5,586,273	0.01%	ChinaNet, China
6	4538	CN	185	95,323	0.19%	CERNET, China
7	4812	CN	93	996,884	0.01%	Chinanet, China

Table 4 – Networks that use Truncated Responses as a proportion of per network test count

It appears that these are isolated cases in most networks where this behaviour is observed.

How reliable is Re-Query using TCP?

Let's now look at the TCP re-query rate.

The results of this analysis are shown in Table 5.

	Dual Stack	V6-Only
Tests	67,469,670	134,295,458
TC	62,471,679	92,581,430
TC-noTCP	562,334	2,612,150
No Query Target	483,557	2,600,142
Rate	0.77%	2.81%

Table 5 – Test of DNS TCP Followup

This is a surprising outcome, where the V6-only outcome is noticeably worse than the dual stack result. The distribution of where these cases occurred in terms of the host network is shown in Table 6 for the top 10 such networks.

Rank	AS	CC	Count	Rate	AS Name
1	55836	IN	582,358	22.31%	Reliance Jio, India
2	45609	IN	305,030	11.69%	Bharti Airtel, India
3	1221	AU	198,111	7.59%	Telstra, Australia
4	4134	CN	154,605	5.92%	Chinanet Backbine, China
5	45669	PK	141,906	5.44%	Mobilink, Pakistan
6	22085	BR	130,686	5.01%	Claro, Brazil
7	9808	CN	107,490	4.12%	China Mobile, China
8	23693	ID	88,869	3.40%	Telekomunikasi Selular, Indonesia
9	30722	IT	85,834	3.29%	Vodaphone-IT, Italy
10	38266	IN	80,317	3.08%	Vodaphone-Idea, India

Table 6 – Networks that contain non-TCP DNS resolvers

Again, is this an issue for all resolvers within these ten networks, or a more isolated set of behaviours?

Rank	AS	CC	no-TCP Count	Total Count	Rate	AS Name
1	55836	IN	582,358	11,803,186	4.93%	Reliance Jio, India
2	45609	IN	305,030	4,334,161	7.04%	Bharti Airtel, India
3	1221	AU	198,111	553,552	35.79%	Telstra, Australia
4	4134	CN	154,605	5,586,273	2.77%	Chinanet Backbine, China
5	45669	PK	141,906	165,545	85.72%	Mobilink, Pakistan
6	22085	BR	130,686	185,642	70.40%	Claro, Brazil
7	9808	CN	107,490	2,720,825	3.95%	China Mobile, China
8	23693	ID	88,869	218,068	40.75%	Telekomunikasi Selular, Indonesia
9	30722	IT	85,834	92,276	93.02%	Vodaphone-IT, Italy
10	38266	IN	80,317	231,838	34.64%	Vodaphone-Idea, India

Table 7 – Networks that use Truncated Responses as a proportion of per network test count

This inability to re-query using TCP appears to be a consistent behaviour for DNS resolvers used in AS30722, Vodaphone Italy, AS45669, Mobilink Pakistan, AS22085, Claro Brazil, AS23693, Telekomunikasi Selular Indonesia and AS1221, Telstra Australia. In these cases, its highly likely that any operational issues would be masked out by the ability to perform these TCP queries over IPv4.

It is highly unlikely that the resolver software used in these networks is incapable of making queries over IPv6, while retaining this ability in IPv4. A more obvious explanation is that the security filter lists that surround these DNS services has only enabled TCP using IPv4 and has omitted an entry for TCP port 53 using IPv6.

Conclusions

Yet again we appear to have encountered a situation using IPv6 with the DNS where we've encountered some operational problems.

It appears that the transition to IPv6 has taken so long that many network service operators have been counting on the dual stack environment to mask over issues that would otherwise be glaringly obvious in a single-protocol stack environment. As long as IPv4 is never turned off then this is not a pressing problem.

The idea behind this dual stack transition and the associated behaviour to prefer to use IPv6 when it is available, is that the more we deploy dual stack infrastructure and services the more we will use IPv6 in preference to IPv4. The use of IPv4 should wane to the point that it is no longer used at all. In this way we can avoid deciding when to "turn off" support for IPv4, as it should disappear as a result of this simple preference rule.

However, if we continue to use the dual stack environment to mask over operational issues with IPv6, notably in the area of packet fragmentation handling and overly restrictive filter settings and such, and implicitly rely on IPv4 to produce the desired service outcome then there will always be a residual level of IPv4 traffic. And if we are after a "natural" end to the transition where there is simply no more IPv4 traffic on the net, then this will just not happen. So, for that reason it is probably worth looking at common services, such as the DNS, in an IPv6-only mode of operation and checking that they really can work correctly without relying on having IPv4 lurking in the background help them out when IPv6 gets stuck!

Otherwise, we might be stuck in this dual stack network for a very very long time!

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