The ISP Column
A column on things Internet

July 2024
Geoff Huston

Revisiting DNS and UDP Truncation

The choice of UDP as the default transport for the DNS was not a completely unqualified success. On the positive side, the stateless query/response model of UDP has been a good fit to the stateless query/response model of DNS transactions between a client and a server. The use of a UDP transport enabled the implementation of highly efficient DNS server engines that managed high peak query rates. On the other hand, these same minimal overheads imply that DNS over UDP cannot perform prompt detection of packet loss and cannot efficiently defend itself against various approaches to tampering with the DNS, such as source address spoofing, payload alteration and third-party packet injection. Perhaps most importantly, the way UDP handles large payloads is a problem.

> Payloads up to 65,507 octets can be loaded into a UDP frame, when using an IPv4 transport. Its 28 octets lower than the IPv4 maximum 16-bit packet length value due to allowing 20 octets for the IPv4 packet header and 8 octets for the UDP header. The maximum payload increases slightly to 65,527 octets when using an IPv6 transport (when not using IPv6 Jumbogram Extension Headers), due to the observation that the 16-bit payload length field in the IPv6 packet header excludes the IPv6 packet header.
>
> In practice, most networks do not cope at all with such large IP packets. IP fragmentation is used to adapt a large IP packet to be passed across a network path that uses a smaller maximum transmission size. With IP packet fragmentation the network only handles packets within its acceptable packet size range, leaving the end systems with the task of working with these large packets. That said, IP fragmentation is still a problem. RFC 8900, "IP Fragmentation Considered Fragile" from September 2020 reiterates advice from a 1987 paper, "Fragmentation Considered Harmful", which points out that that relying on the IP layer to perform the necessary adaptation to accommodate large payloads in a single datagram transaction is a very poor approach from the perspective of carriage performance. Furthermore, a current work-in-progress in the IETF, "IP Fragmentation Avoidance in DNS over UDP", points out that fragmented DNS responses have systematic weakness that expose a DNS requestor to DNS cache poisoning from off-path attackers. As this work points out: "A DNS message receiver cannot trust fragmented UDP datagrams primarily due to the small amount of entropy provided by UDP port numbers and DNS message identifiers, each of which being only 16 bits in size, and both likely being in the first fragment of a packet if fragmentation occurs.

The DNS avoids IP fragmentation by restricting the maximum payload size carried over UDP. RFC 1035 contains the directive: "Messages carried by UDP are restricted to 512 octets (not counting the IP or UDP headers). Longer messages are truncated, and the TC bit is set in the header."

The intent of setting the TC bit in the DNS response was to indicate that the receiver should discard this UDP response and perform the same DNS query over a TCP transport. This is not exactly a highly efficient measure. The DNS query now takes an additional 2 round-trip time intervals (one for the DNS truncated response and a further exchange for the TCP handshake), and the server also needs to maintain a TCP state, which limits the server's query processing capability. This option to requery using TCP is preferably avoided but limiting DNS responses to at most 512 octets is not always feasible.

This limit of 512 octets becomes problematical in a number of scenarios. For example, a DNSSEC-signed query for the DNSSEC public keys of the root zone produces a response of 1,169 octets. It's not just DNSSEC that is the issue here. We use the DNS for various form of authentication, and it's a common practice to load authentication codes into the DNS as TXT records. Multiple TXT records will all be packed into a response, which can lead to quite large responses. For example, a query for the TXT record for `bbc.co.uk` elicits an DNS response of 1,666 octets in size.

The workaround for this issue of a very conservative selection of the maximum UDP payload for the DNS was the use of a so-called pseudo–Resource Record, the OPT record. This is the general extension mechanism for DNS, or *EDNS*. The specification for EDNS(0), RFC 6891, includes the option to use a DNS message size in the query to allow the querier to inform the responder of its capability to handle DNS over UDP responses greater than 512 octets, thereby avoiding some level of requerying over TCP when the response is larger than this default size. RFC 6891 also contains the following advice: "Due to transaction overhead, it is not recommended to advertise an architectural limit as a maximum UDP payload size. … A good compromise may be the use of an EDNS maximum payload size of 4096 octets as a starting point."

The IPv6 specification requires IPv6 networks and hosts to be capable of handling an IPv6 packet of up to 1,280 octets in size without fragmentation. The IPv4 specification has an unfragmented packet size of 68 octets, and IPv4 hosts are required to be capable of reassembling IP packets of up to 576 octets in length. In practice, the original Ethernet packet sizes (64 to 1,500 octets) have been largely adopted by the Internet, and in most cases (where no encapsulation tunnels exist) packets of up to 1,500 octets will pass through the public Internet without triggering packet fragmentation. What this implies is that in proposing a UDP buffer size of 4,096 octets, then IP fragmentation of large DNS responses will be triggered, and all the issues raised relating to the use of UDP fragments may surface as a consequence.

If a primary objective is to avoid IP packet fragmentation, then a UDP buffer size of 4,096 octets is just too large. The advice in DNS Flag Day 2020 proposed the use of an EDNS(0) buffer size of 1,232 octets as a minimum safe size, based on the 1,280 octet unfragmented IPv6 packets, and making allowance for the IPv6 and UDP packet headers. However, this is a very conservative choice, and the downside is potentially unnecessary requeries in TCP.

A current work in progress in the IETF, draft-dnsop-avoid-fragmentation proposes that the EDNS buffer size should reflect not only the requestor's maximum packet reassembly buffer size, but also the inbound network interface MTU, and where known, the network path MTU value. This working draft currently recommends a maximum buffer size of 1,400 octets.

Measuring EDNS Buffer Sizes

Which brings us to our first measurement question. To what extent do recursive resolvers follow this advice? What are the UDP buffer sizes used in queries from recursive resolvers to name servers?

We have looked at the UDP Buffer size in queries in June 2024, as shown in Table 1 and a cumulative distribution of this data is plotted in Figure 1. This is a query-weighted data set, using data from some 356,939,321 query sample points, collected over nine days from the 25th June 2024 to the 3rd July 2024.

Table 1 shows the top 10 buffer sizes used in queries.

Size	Query Count	%
1400	197,925,500	55.45%
4096	65,859,104	18.45%
1232	57,497,614	16.11%
1220	14,550,191	4.08%
1452	7,296,654	2.04%
512	4,986,122	1.40%
1410	2,343,952	0.66%
1472	2,187,606	0.61%
1440	1,495,692	0.42%
1680	594,826	0.17%

Table 1 – Top 10 UDP Buffer Sizes

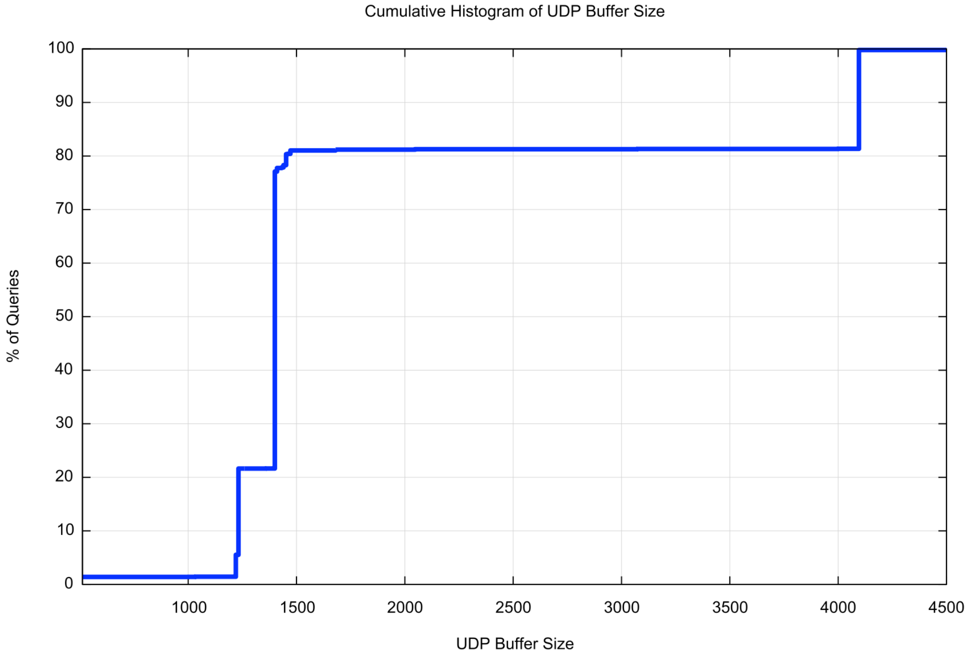

Figure 1 – Distribution of UDP Buffer Size values in DNS Queries

Just under 20% of these queries use a UDP buffer size greater than 1,472, which would appear to allow a responder to generate a fragmented UDP response, unless of course it applies its own more stringent size restraints to the UDP_response.

Measuring Truncated Responses

The next question is: "How effective is truncation in today's DNS?"

Do DNS recursive resolvers always ignore the answer and additional sections of a DNS response if the truncated bit is set? And what proportion of resolvers are capable of performing a DNS query over UDP in response to a truncated UDP response?

We've used the APNIC Labs Ad-based measurement environment to perform this measurement.

This system uses online ads to enrol users to test particular behaviours from the perspective of the end user. The script in the ad performs a number of fetches of URLs. Each URL to be fetched uses a unique DNS name, and there is only a single authoritative DNS server to resolve this name, and a single web server to serve the named web object. We cannot instrument the end user browser that is running the ad script, but we can instrument the DNS and Web servers to record their end of each measurement transaction. Each fetch within as single ad script can be used to measure a particular behaviour or attribute, such as IPv6-only capability, use of DNSSEC to validate domain name resolution, the use of QUIC, and the adoption of network behaviours that drop routes have invalid ROAs. The only control framework of the script on the user side measures the elapsed time to perform each URL fetch and passes this information back to an ad controller by performing a final URL fetch with the individual time values encoded as attributes to this closing report.

The ad system is configured to present some 15M – 20M ad impressions per day, with an ad configuration that attempts to secure as wide a diversity of end users as possible. On the server side we use a distributed network of five server-side platform clusters, located approximately on each continent to try and minimise the network delays when each user connects to the experiment's DNS and Web servers.

To perform this measurement of DNS resolver handling of truncated responses and the related ability to switch to use TCP we've used a *glueless* DNS technique. This allows us to use the DNS itself to detect whether a DNS resolution environment can resolve a DNS name that is constructed using a particular DNS behaviour.

Generically, the technique is to use a target DNS name that is itself a delegated DNS name, and in the DNS delegation data the glue records are deliberately omitted. This is shown in Figure 2.

```
example.com zone
child NS ns1.oob.example.net.

oob.example.net zone
ns1  IN   A   192.0.2.1

child.example.com zone
.    IN    A   203.0.113.1
```

Figure 2 – Example of Glueless Delegation

In this example, to resolve the DNS name `child.example.com`, the recursive resolver will ask the name server for the `example.com` zone, and the server will response with a referral record, indicating that the name is defined in the zone `child.example.com`, and the name server for that zone is `ns1.oob.example.net`. However, the referral response does not contain any glue records to provide an IP address for this name, so the recursive resolver must set aside its primary task (resolving `child.example.com`) and start a new task to resolve the name `ns1.oob.example.net`.

If it is successful, the resolver now has an IP address for the name server of the original target zone, and it can ask the final question. It will only ask this final query if the resolution of the nameserver name is successful.

In this case we have modified the behaviour of the DNS server for the second zone (`oob.example.net`), such that all UDP responses to queries for name server names in this zone are truncated. We also use a categorization of these names such one half of experiment's unique name set causes the nameserver to generate a truncated response (TC=1) with an empty answer section and the other half of the query names generate a perfectly normal complete DNS response with an intact and complete answer section, but with the TC bit set indicating (incorrectly in this case) that the UDP response has been truncated.

If the DNS resolver is using the contents of a truncated UDP response, then it will be able to obtain the IP address of the nameserver and make the third query without needing to requery using TCP. A standards-compliant resolver will ignore the answer section of the UDP response that had the truncated bit set and requery using TCP and use the TCP response to make the third query.

The behaviours are determined by performing a full recording of all packets that arrive at and leave our servers, and then analysing these packets to determine the DNS resolution query and response sequences for each individual experiment.

In a test conducted over June 2024 we found results as shown in Table 2.

Tests	394,766,935
Ans+TC	197,173,501
No TCP	96,401
Rate	0.05%

Table 2 – Incidence of Use of Answer Section in Truncated Responses

Across some 394 million sample points, some 197 million tests were provided with a complete answer section as well as having the truncated bit set. Of these some 96,401 tests did not requery using TCP, but performed the third target query, evidently using the contents of the answer section in the truncated response. The ten network with the highest proportional use of truncated answers is shown in Table 3.

ASN	CC	Samples	Fail Rate	AS Name	Country
30549	CA	431	60.56%	LAKELAND-NETWORKS	Canada
36923	NG	1,045	54.55%	SWIFTNG-ASN	Nigeria
17882	MN	127,498	29.52%	UNIVISION	Mongolia
16509	IN	4,761	23.61%	AMAZON-02	United States (India)
26421	US	219	11.42%	PONDEROSA-INTERNET	United States
17816	CN	169,488	9.42%	China Unicom Guangdong Province	China
42455	IM	122	8.20%	WI-MANX-AS	Isle of Man
16284	NG	334	4.79%	Inq-Digital	Nigeria
61272	LT	328	4.27%	IST	Lithuania
16509	DE	4,815	3.74%	AMAZON-02	United States (Germany)

Table 3 – Incidence of Use of Answer Section in Truncated Responses – Top 10

These results suggest that globally this aspect of DNS conformance to standards-specified behaviour is not a severe problem, and the incidence of the use of the answer section contained in truncated responses is just 0.05% of all samples. However, as shown in Table 3, the incidence of this DNS resolver behaviour in specific networks is not so small, and this table lists the 10 networks with the highest incidence of the use of truncated responses where more than 100 samples were gathered over the month of June 2024.

Measuring TCP Requery

The second part of a resolver's actions when receiving a DNS response over UDP that has the truncated bit set is to requery using TCP. The related measurement question is: What proportion of resolvers are incapable of performing a DNS query over TCP?

The overall results are shown in Table 4.

There is a visible level of use of TCP-only here. Some 439,900 tests performed the DNS resolution by asking the initial query over TCP rather than UDP. This represents 0.11% of the total count of 394 million tests. The remaining tests were initiated over UDP and given a truncated response.

Tests	394,766,935
TCP only	439,900
Rate	0.11%
TC+UDP	394,327,035
UDP+NO TCP	10,555,279
Rate	2.67%

Table 4 – TCP Use Profile

Of these 394 million tests, some 10.5 million users performed the initial query over UDP, received a truncated response, and then failed to requery using TCP. This represents 2.67% of all such tests. The 10 largest TCP failure rates for networks with at least 500 sample points are shown in Table 5.

This 3% failure rate is larger than the 0.05% of users who use the answer section of truncated responses, but even a 3% failure rate is not a major issue for the DNS when seen as a network-wide system. However, there are individual networks, both large and small, where there is a far higher TCP failure rate. Such high failure rates, in excess of 90% of tests for users within each of these networks, suggest that the issue is likely to lie in the DNS resolver infrastructure operated by these networks rather than end clients performing their own DNS recursive resolution functions.

AS	CC	Samples	No-TCP Rate	Country Name	Country
9444	HK	126	98.41%	Hong Kong Telecommunication	Hong Kong
22354	TZ	702	96.72%	University of Dar es Salaam	Tanzania
41937	RS	25,475	95.43%	MOJASUPERNOVA	Serbia
51357	UA	239	94.98%	SMARTCOM	Ukraine
37229	TG	8,087	94.94%	Atlantique Telecom	Togo
10396	PR	339,523	93.27%	COQUI-NET	Puerto Rico
16116	IL	89,859	92.90%	Pelephone Communications	Israel
272744	BR	239	91.63%	DC INTERNET EIRELI	Brazil
6535	CL	129,710	91.00%	Telmex Servicios	Chile
38819	HK	110,128	90.11%	HKCSL GPRS	Hong Kong

Table 5 – Incidence of no TCP followup to Truncated UDP Responses – Top 10

Conclusions

The DNS is attempting to steer a careful path between the issues associated with response loss associated with UDP and packet fragmentation and response loss with truncation and requerying with TCP. Previous measurements of fragmented UDP failures rates in the DNS between recursive resolvers and authoritative nameservers showed a fragmented response failure rate of around 15% (https://www.potaroo.net/ispcol/2020-12/xldns2.html), while the current failure rate of truncation and TCP is far lower at some 3%. On the basis of preferring the lesser of the potential loss rates, an approach of using a lower maximum UDP size that avoids fragmentation in favour of requerying in TCP appears to represent a means of improved robustness when handling large DNS responses.

This measurement exercise does not attempt to identify individual recursive resolvers. Modern high-capacity recursive resolver systems are compound systems composed of a number of DNS resolution engines. Some DNS resolver engines may use only UDP, where TCP tasks may be handed to other resolver engines that are specifically configured to manage the somewhat different TCP load profile. Without undertaking an effort to identify the modes of behaviour of these compound systems, identifying individual resolver systems by their IP address is not overly useful when trying to identify systemic behaviour anomalies.

A report on DNS Requery TCP failure rates can be found at https://stats.labs.apnic.net/dnstcp.

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