Fragmentation, truncation, and timeouts: are large DNS messages falling to bits?

SIDN Labs Tech Report #2020-03
2020-11-20

Giovane C. M. Moura[1], Moritz Müller[1,2], Marco Davids[1], Maarten Wullink[1], and Cristian Hesselman[1,2]

[1] SIDN Labs, Arnhem, The Netherlands
[2] University of Twente, Enschede, The Netherlands
{firstname}.{lastname}@sidn.nl

Abstract. The DNS provides one of the core services of the Internet, mapping applications and services to hosts. DNS employs both UDP and TCP as a transport protocol, and currently most DNS queries are sent over UDP. The problem with UDP is that large responses run the risk of not arriving a their destinations – which can ultimately lead to *unreachability*. However, it remains unclear how much of a problem these large DNS responses over UDP are in the wild. This is the focus on this paper: we analyze 114 billion queries/response pairs from more than 43k autonomous systems, covering two months and a week period (2019 and 2020), collected at the authoritative servers of the `.nl`, the country-code top-level domain of the Netherlands. We show that fragmentation, and the problems that can follow fragmentation, rarely occur at such authoritative servers. Further, we demonstrate that DNS built-in defenses – use of truncation, EDNS0 buffer sizes, reduced responses and TCP fall back – are effective to reduce fragmentation. Last, we measure the uptake of the DNS flag day in 2020.

1 Introduction

The Domain Name System (DNS) [27] provides one of the core Internet services, by mapping hosts, services and applications to IP addresses. DNS specifications states that both UDP and TCP should be supported [27,4] as transport protocols, and nowadays most queries use UDP. Performance wise, UDP's main advantage is that it can deliver faster responses, within one round-trip time (RTT), while TCP responses require an additional RTT due to its handshake.

Often common, small DNS responses may fit into the 512-byte limit that the original DNS over UDP (DNS/UDP hereafter) has, but larger responses – such as the ones protected with DNSSEC [3,23,4] – do not fit. To overcome this 512-byte limit, the Extension Mechanisms for DNS 0 (EDNS0) [48,7] was introduced. It allows a DNS client to advertise its UDP buffer size, and an EDNS0-compatible authoritative server "may send UDP packets up to that client's announced buffer size without truncation" [48] – in theory up to 65,536 bytes.

If, however, the designated response is larger than the client's advertised EDNS0 limit (or 512 bytes in the absence of EDNS0), the authoritative server should then *truncate* it to a size that fits within the limit and flag it [28]. Upon receiving a truncated response, the client should, in turn, resend the query over TCP [10,4] (DNS/TCP hereafter), and leverage TCP's design to handle large messages with multiple segments.

However, the EDNS0 announced buffer size is agnostic to the path between client and authoritative server's maximum transmission unit (MTU), which is the largest packet size that can be forwarded by all routers in the path. The most common MTU on the core Internet is 1500 bytes [4], and EDNS0 buffer sizes can easily exceed – we show in §4 that 4096 bytes is the most common value. If it does *exceed* the entire path MTU, then the packet will *not* be able to be forwarded by the routers, which may lead to packets being discarded or being *fragmented* [35,11] at the IP layer.

IP fragmentation, in turn, comes with a series of problems [5] – fragmented IP packets may be blocked by firewalls [8,4,5], leading to *unreachability* [46,49]. Moreover, IP fragmentation has been exploited in cache poisoning attacks on DNS [16,45], and DNS cache poisoning can be further exploited to compromise the trust in certificate authorities (CAs) [6]. As as result of these problems, there is currently a consensus in the IP and DNS communities that IP fragmentation should be avoided in DNS [5,12,53].

In this paper, we scrutinize the issue of large DNS responses using as vantage point the `.nl` zone, the country-code top-level domain of the Netherlands. Our datasets cover 2 months and 1 week of data, from 2019 and 2020, with more than 114 billion queries/responses pairs from more than 3 million resolvers from more than 45,000 Autonomous Systems (ASes). We investigate responses sizes, truncation, and server-side fragmentation in §3, as well as determining if resolvers fall back to TCP. Then, in §4, we characterize resolver's EDNS0 buffer sizes and the uptake of the DNS Flag day 2020.

2 Datasets

There are two main types of DNS server software: *authoritative servers* and *recursive resolvers*. Authoritative servers "know the content of a DNS zone from local knowledge" [17] (such as the Root DNS servers [41] for the Root zone [19]), while DNS resolvers (such as the Quad{1,8,9} public resolver services [15,36,1,32]), resolve domain names by querying authoritative servers on behalf of users.

We analyze DNS queries and responses to/from authoritative servers of `.nl`, the country-code top-level domain (ccTLD) of the Netherlands. We collect data from two of the three authoritative server of `.nl` (NS1 and NS3, the remaining authoritative services did not support traffic collection at the time). The `.nl` zone has several million domain names in its zone, with the majority of the domains being signed using DNSSEC [43].

The analyzed authoritative servers are run by different third-party DNS providers (one from Europe, the other from North America). Both services are replicated using IP anycast [25,33] – which allows the same IP address to be announced using BGP [37] from multiple locations across the globe, over both IPv4 and IPv6. In total, NS1 and NS3 are announced from 61 global locations (sites). We employ ENTRADA [42,51], an open-source DNS analysis platform to analyze this data.

Table 1 shows the datasets we analyze in this paper. In total, we study more than 114 billion DNS queries and responses – 109 billion over UDP and 4.41 billion over TCP, covering two full months (July 2019 and 2020) and the first week of October 2020 (the first week *after* the DNS 2020 flag day [53]).

	July 2019		**July 2020**		**Oct. 2020 (1–7)**	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
Queries/responses	29.79B	7.80B	45.38B	15.87B	11.17B	3.97B
UDP	28.68B	7.54 B	43.75B	15.01B	11.10B	3.77B
UDP TC off	27.80B	7.24B	42.06B	13.88B	10.76B	3.54B
UDP TC on	0.87B	0.31B	1.69B	1.14B	0.34B	0.23B
Ratio (%)	2.93%	3.91%	3.72%	7.15%	2.99%	5.72%
TCP	1.11B	0.25B	1.63B	0.85B	0.36B	0.20B
Ratio (%)	3.72%	3.32%	3.59%	5.37%	3.17%	5.09%
Resolvers						
UDP TC off	3.09M	0.35M	2.99M	0.67M	1.89M	0.27M
UDP TC on	0.61M	0.08M	0.85M	0.12M	0.58M	0.09M
TCP	0.61M	0.08M	0.83M	0.12M	0.58M	0.09M
ASes						
UDP TC off	44.8k	8.3k	45.6k	8.5k	42.9k	7.9k
UDP TC on.	23.3k	4.5k	27.6k	5.4k	26.6k	5.0k
TCP	23.5k	4.3k	27.3k	5.2k	24.3k	4.8k

Table 1: Evaluated datasets of `.nl` zone.

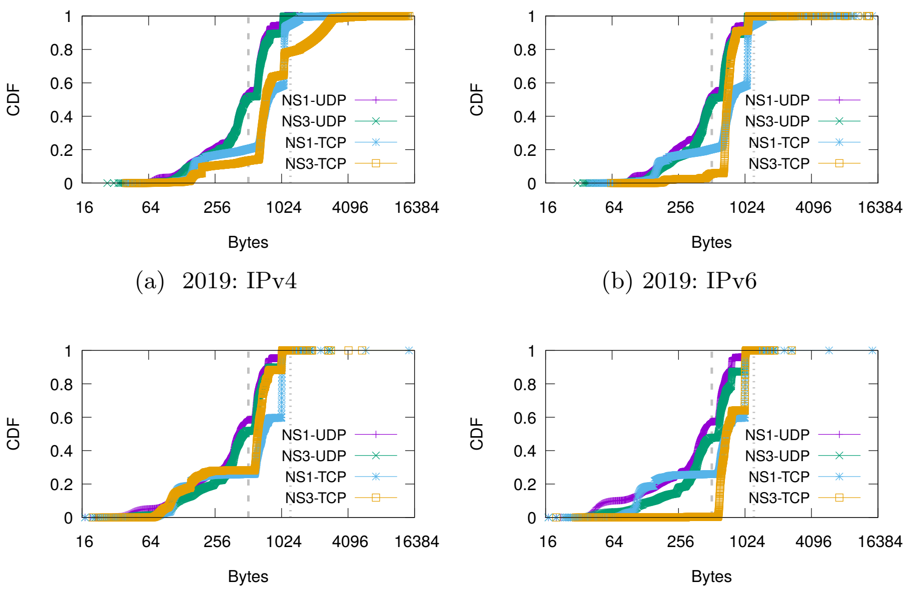
We see that a small fraction of all responses are truncated – 2.93% to 7.15% – depending on the month/year and IP version. Our datasets cover more than 3 million resolvers (defined by distinct IP addresses) from more than 45k ASes, which is far larger than previous studies on DNS issues with fragmentation [49,46] and from active measurements platforms such as Ripe Atlas [40], which has ∼11k active vantage points and cover 8670 /24 IPv4 network prefixes [39] (May 2020).

3 Dissecting Responses from a ccTLD

3.1 How common are large responses?

Before addressing problems related to large DNS/UDP responses, we need first to understand how often do they really occur in the wild, using our `.nl` vantage point.

Figure 1 shows the CDF of the response sizes (DNS payload only) per anycast server, transport protocol, and IP version, for both July 2019 and July 2020. We see that most responses are smaller than 1232 bytes (right vertical line) – more than 99.99% for all responses, for both servers, protocols/IP version.



(a) 2019: IPv4 (b) 2019: IPv6

(c) 2020: IPv4 (d) 2020: IPv6

Fig. 1: Response size CDF for `.nl`: July 2019 and 2020

This value is similar to what is reported by Google Public DNS [15], a public DNS resolver service, also reports that 99.7% of responses are smaller than 1232 bytes [24]. Differently from ours, they run a *resolver* service, that queries *multiple* TLDs and their delegations, while ours covers only one ccTLD. Still, similar figures holds for both vantage points.

The exception for `.nl` was in 2019, where NS3-TCP over IPv4 had 78.6%, and NS1-TCP over IPv6 had 94.9% of the responses smaller than 1232 bytes. Altogether, for July 2019 and 2020, these large responses account for 95M queries, out of the more than 98B queries (Table 1).

What queries generate large responses? We then proceed to determine what queries caused large responses. DNSSEC is often blamed for causing large responses. At `.nl`, DNSSEC definitely increases response size, but rarely beyond 1232 bytes.

Resolvers set the DO-flag in their queries if they want to receive DNSSEC related resource records for each signed response (e.g. DS and RRSIG). Responses to these queries have a median response size of 594 bytes, whereas responses that do not contain DNSSEC records only have a median response size of 153 bytes. Responses that stand out are A [28] and AAAA [44] queries (asking for IPv4 and IPv6 records, respectively) for `ns*..dns.nl` – the authoritative servers of the `.nl` zone, accounting for 99% of all responses larger than 1232 bytes. Without DNSSEC records, this response is merely 221 bytes long.

We further found that the responses sizes for these queries *changed* per authoritative service. For NS1, the responses were 217 bytes long (median), but responses from NS3 were 1117 bytes long.

This staggering difference is due to configuration differences between the servers. NS1 is configured to return minimal responses [20,2], and its responses do not include two sections with "extra" records (authority and additional records section [27]). The NS3 operator, did not enable this feature, which inflates response sizes. That shows us that not only the query type is important, but how minimal responses influences response sizes and truncation.

3.2 How often does IP fragmentation occur for DNS/UDP?

IP fragmentation can take place either at the authoritative servers (for both IPv4 and IPv6) and on the routers along the way only for IPv4, but only if the IP Don't Fragment flag (DF) in the IPv4 is not set. For IPv6, fragmentation only occurs on the end hosts (§5 in [9]).

Server-side fragmentation: If a DNS/UDP response is *larger* than the authoritative server's link MTU (and the server is not limited from large responses (`max-udp-size` in BIND9 [20]) the the server may fragment it.

Given we do not run NS1 and NS3, we cannot know what is their `max-udp-size` limits. What we can know, however, is what is the *largest* DNS/UDP response they have sent and that was not fragmented. This value provides a lower bound for their `max-udp-size` of the authoritative servers. Table 2 shows the results. We see that in NS3 send far larger responses than NS1 in 2020.

Then, we proceed to analyze the number of DNS/UDP fragmented responses per authoritative server and IP version. Figure 2 shows a timeseries of these responses. We see very few occur: fewer than 10k/day, compared to a total of 2.2B/day. Notice that NS1 has no fragmented responses in 2020, which is probably due to the reduction on the response sizes in 2020 (Table 2).

Still, even if there are few fragmented queries, why do they still occur? First, we see most fragmented queries are from NS3 (Figure 2), given NS3 does not return minimal responses (§3.1), which inflates responses. This shows that minimal responses are an effective method to reduce fragmentation.

	NS1		NS3	
Year	*IPv4*	*IPv6*	*IPv4*	*IPv6*
July 2019	1451	1470	1484	1494
July 2020	1391	1391	2866	2866

Table 2: Maximum DNS/UDP response size (bytes) per authoritative server and IP version , disregarding IP/UDP headers.

	IPv4	IPv6
	ICMP Type3, Code4	*ICMPv6 Type 2*
July 2019	73	16
July 2020	641	599

Table 3: NS3 - ICMP error messages caused by large packets.

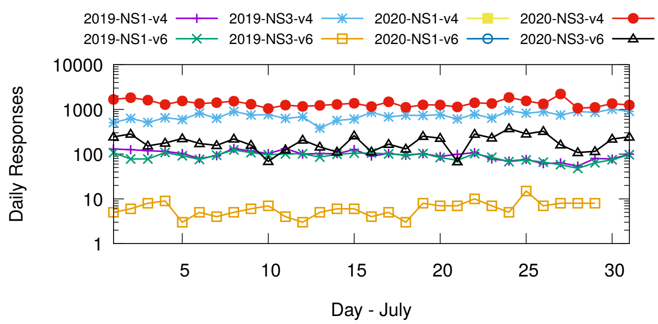


Fig. 2: UDP fragmented queries for `.nl` authoritative servers.

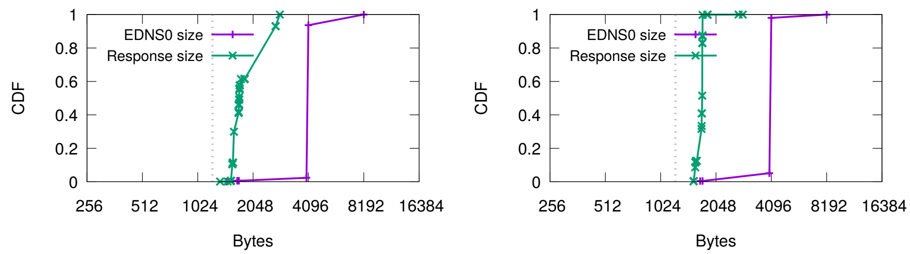
But the resolvers have their own share of responsibility. We single out these DNS/UDP fragmented responses, and analyzed the announced EDNS0 buffer sizes. Figure 3 shows the results for July 2020, for both IPv4 and IPv6. We see that most fragmented queries are smaller than 2048 bytes, but we see that most of these resolvers announced a large EDNS0 buffer size – most equal to 4096 bytes, which is default value on BIND (up to version 9.16.6)[34] [20].

Packets larger than path MTU: Since we collect traffic only at the authoritative servers, we cannot directly know if there was IPv4 fragmentation along the path. However, we can still use the ICMP protocol to determine if *some* of the DNS responses exceed the path MTU.

The routers along the path have a standard way of handling IP packets larger that their MTU, both using ICMP. If it is an IPv4 packet, and the fragmented flag (DF) is set, then the router should discard the packet and send a ICMP Type 3, code 4 packet as a response ("Fragmentation Needed and Don't Fragment was Set" [34]) back to the authoritative server. If the DF flag is off, then the router can fragment the packet – and no ICMP signaling is sent back to

[3] BIND9 uses a *dynamic* EDNS value: when it first contacts a server, it uses 512 bytes. From that point on, it uses the configured value – 4096 by default. If it receives no responses, it will lower it to 1432, 1232 and 512 bytes. See `edns-udp-size` in [20].

[4] Unbound version 1.12.0 set the value to 1232, on October 8th 2020 [50], and so did BIND on version 9.16.8.



(a) 2020: IPv4 (b) 2020: IPv6

Fig. 3: Fragmented Queries July 2020: response sizes and EDNS0 buffer sizes.

the authoritative server. Last, IPv6 packets cannot be fragmented by routers, and routers facing them should send an ICMPv6 Type 2 message ("packet too big" [22]) back to the authoritative server.

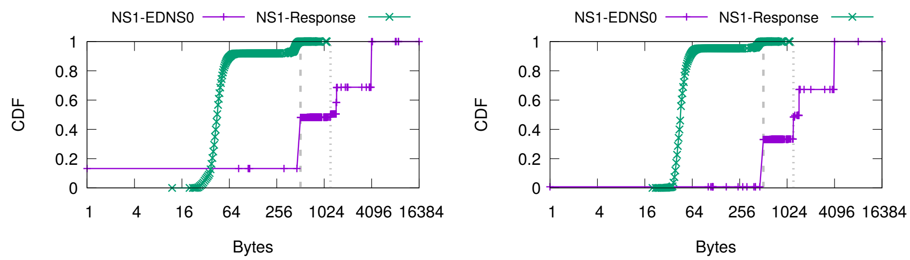
In our setup, only the DNS provider of NS3 provides us with ICMP traffic. We analyze the ICMP traffic and show in Table 3 distribution of ICMP error messages associated with large packets, and there are only few of them .

In the worst case scenario, these large DNS/UDP would be discarded by routers and both client and servers would not know about it, which could, in theory lead to unreachability. However, previous research has shown that, in the wild, DNS resolvers have built-in a series of fail-tolerance features, and will retry *multiple times* the same server and or switch from server/IP version, to the point of "hammering" the authoritative severs, in order to obtain responses [29,31]. In this scenario, even if one authoritative server becomes "unresponsive" – from the point-of-view of the resolver – having multiple authoritative servers (defined by distinct NS records), running on *dissimilar* networks, should minimize the probabilities of unreachability.

Network issues with large responses: our vantage point does not allow to know if clients received their large DNS/UDP responses. To asses that, we then resort to Ripe Atlas probes and NS3, and evaluate 1M queries from roughly 8500 probes, over a period of one day. We show in §B that 2.5% of small (221 bytes) DNS/UDP responses, do not make it to clients. For large responses (1744 bytes), this value is 6.9% – only considering a single DNS/UDP query without TCP fallback. Comparing to server-side fragmentation, we show that it is far more likely to happen on the network. Similar numbers were reported by Huston [18], who measured 7% drop with a similar response size on IPv6 and Van den Broek et al. [47] have shown that even up to 10% of all resolvers might be unable to handle fragments.

3.3 DNS truncation: how and when?

Table 1 shows that 2.93–7.15% of all evaluated queries were truncated. Next we investigate why this happens. For each truncated response, we fetch its response size and its respective query' EDNS0 buffer size. Figure 4 shows the CDF for these values for July 2020, for NS1 (§A shows NS3 for 2020 and the 2019 results for NS1 and NS3). We see that most DNS/UDP responses are truncated to values under 512 bytes, independently IP version (Response line).



(a) NS1: IPv4 (b) NS1: IPv6

Fig. 4: NS1: CDF of DNS/UDP TC responses for `.nl`: July 2020

Small or no EDNS0 values lead to truncation: we see that most EDNS buffer sizes are equal to 512, which is rather too small for many queries (but the initial value by BIND when it first contact a server [20]). As such, if resolvers would advertise larger buffers, that would probably reduce truncated responses.

Oddly, we also see that only NS1 receives 13% of queries that are truncated with no EDNS0 extension, but not the other servers or IP version (shown as EDNS0=1 in Figure 4). We found that this is due to an anomaly from two ASes (AS2637 – Georgia Tech and AS61207 – Ilait AB). Resolvers from these ASes have a "sticky" behavior [31], sending queries only to NS1 over IPv4. Both ASes send most queries without EDNS0 UDP buffer value (1 in the graph), and that is why Figure 4a is skewed.

Large EDNS0 values is no insurance against truncation: We also see that even if clients announce large EDNS0 buffers, they still receive truncated responses. Even though 4096 bytes is enough to fit most responses (§3.1) the server's local MTU is an effective upper limit.

3.4 Do resolvers fall back to TCP?

Upon receiving a DNS/UDP truncated response, DNS resolvers *should* resend the query over TCP – what is know as *TCP fall back* [10]. In July 2020 (Table 1), we see 7.15% DNS/UDP TC queries over IPv6. However, we see only 5.37% of TCP queries over IPv6 – suggesting 1.78% were not followed by DNS/TCP queries. We next investigate this behavior.

Figure 5 shows how many replies with TC flag are followed by a retry via TCP after 60 seconds. The majority, 80% in IPv4 and 75% in IPv6 of these replies are retried via TCP within this time frame per day in July 2020 (on

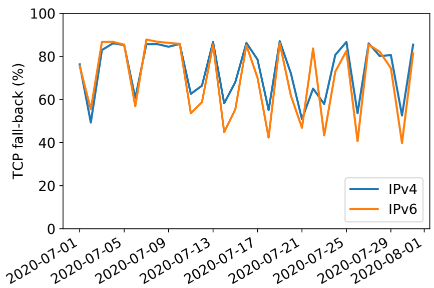


Fig. 5: TC replies with TCP retries

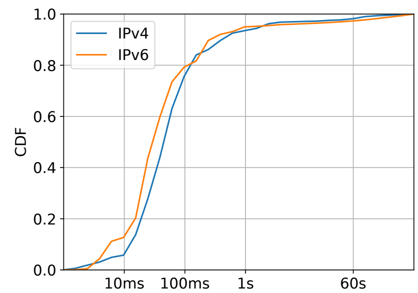


Fig. 6: Time until first TCP fall back

median). For zones where responses often are larger than 1232 bytes this means that after the Flag Day, they will see an increase in TCP connections.

If a resolver retries a query via TCP, then this query is sent usually within less than 100 ms. Figure 6 shows the time between the name server received the initial UDP query and the TCP retry on July 1 2020. 80% of all retries are sent within 100 ms and 90% within one 1 s. Retries from IPv6 addresses reach our authoritative servers slightly faster.

Missing TCP queries: there are multiple reasons why truncated queries may not be followed by TCP ones. For example, queries from non resolvers (such as `dig`) or bots/malware may not comply to that. Also, as we discuss in §2, our datasets do not include data from NS2, the other anycast authoritative server for `.nl`. Given resolvers may switch from server to server [31], our dataset misses those[5]. Resolver farms may be partially to blame – the TCP query may be sent from adjacent IP addresses[6]. Dual-stacked resolvers may only send a TCP query over one (the first) IP version response arriving[7]. Altogether, we estimate that we miss up to 4.8% of retries in our initial measurement. This, still leaves 15–21% of TC replies without a TCP retry. We found that, on July 1st 2020, 39% of these queries were from Google (AS15169), a large resolver operator with multiple levels of caching [30].

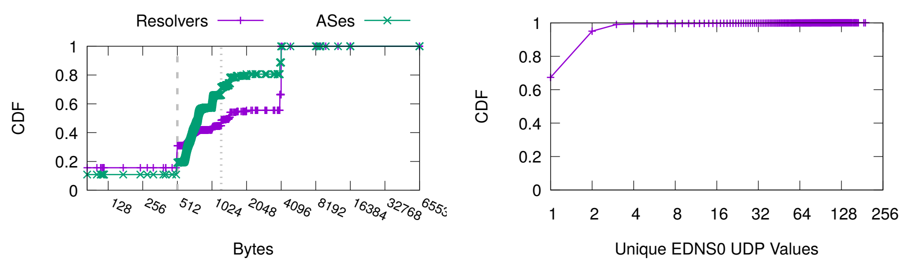
[5] We see 1.9% of TC IPv4 queries switching between NS1 and NS3 on July 1st, 2020, and 3.2% of IPv6 TC queries.

[6] For July 1 2020, we measure, how many TCP retries are first issued from a different resolver than the resolver of the original UDP query, but located in the same subnet (/24 subnet for IPv4 and /48 subnet for IPv6). There, 1.6% of retries via IPv4 and 0.1% via IPv6 are sent from a different resolver, likely belonging to the same farm.

[7] Of a sample of 3M queries that trigger a TC response, 4% were likely issued by those kind of resolvers. 58% then sent their TCP retry via both interfaces, leaving 42% of the TC replies without a TCP retry. Extrapolating these numbers to our measurements we can assume that around 1.3% of TC replies are not retried via TCP because of dual stacked resolvers

4 Resolver EDNS0 bufffer sizes

We now analyze the EDNS0 buffer size for all resolvers we seen in our datsets (Table 1). For 2020, we see in Figure 7a that roughly 30% of all resolvers announce 512 bytes EDNS0 buffer sizes or less, and 48.86% announce 1232 or less. The majority announce 4096 bytes: 33%. For ASes, we have a more even distribution: 20% announce 512 bytes or less, and 71% announce up to 1232 or less. Taking altogether, we can conclude that most resolvers announce a 4096 ENDS0 buffer size value, which is BIND9 default value up to version 9.16.7, is partially to blame for DNS/UDP truncation and fragmentation.



(a) EDNSS0 Values distribution (b) Unique EDNS per resolver

Fig. 7: EDNS0 per resolver and values: July 2020

Figure 7b shows he number of unique EDNS0 buffer sizes announced per resolver for the month of July 2020. We can see that more than 60% of resolvers announce one EDNS0 value over the period (7%, not shown in the figure, have no EDNS0 support) – maybe from old clients and/or non fully compliant software. Only 5% of the resolvers showed 3 or more EDNS0 values in the period.

4.1 DNS Flag Day 2020: what was the uptake?

The DNS Flag Day 2020 was proposed by members of the DNS community in order to avoid IP fragmentation on DNS/UDP, by not allowing UDP queries larger than 1232 bytes. This value was chosen based on a MTU of 1280 bytes – the minimum required by IPv6 [9] – minus 48 bytes of IPv6/UDP headers. The chosen date (2020-10-01) was a suggestion for operators to change their authoritative DNS servers and DNS resolvers.

To measure the *resolvers* uptake, we compare the EDNS0 buffer size values from resolvers from July 2020 to the first week of October 2020, from Table 1. The former we used it as a baseline for comparison, and the latter covers the first week *after* the Flag Day, which we determine the 1232 ENDS0 adoption from the resolvers size. To do that, for each resolver on each dataset, we extract its announced EDNS0 values, and compare them in to see if there were any changes

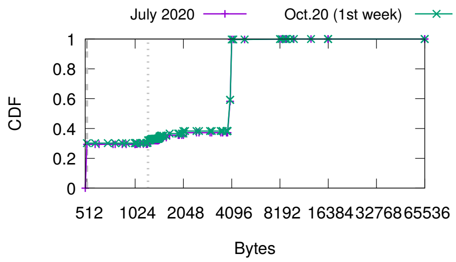


Fig. 8: CDF EDNS0 resolvers

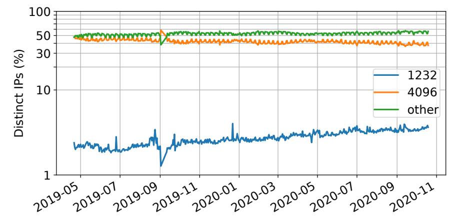


Fig. 9: Daily EDNS buffer distribution by resolvers (y axis in log-2 scale).

from July 2020 to October 2020. Table 4a summarizes this data. We see in total 1.58M resolvers active on both datasets, and they sent 69B queries in the period.

	July 2020	**Oct 1-7,2020**
Resolvers	3.78M	2.24M
∩	1.58M	
UDP Queries	60.3B	15.1B
∩	69.0B	

(a) Before and After Datasets

Resolvers	**6126**
from 4096 bytes	3630
from 1680 bytes	1723
from 512 bytes	565
rest	208
ASes	629

(b) EDNS0 1232 resolvers

Table 4: DNS Flag Day datasets and Changing Resolvers

Figure 8 shows the CDF of resolvers' EDNS0 buffer sizes. We see hardly any changes in the resolver EDNS behavior (if the resolver had multiple EDNS values, we picked the most frequent, also to remove BIND9 512 byte at the first try). On July 2020, we see 30.9% of the resolvers using EDNS0 buffers smaller or equal to 1232 bytes, and on October 2020, this value went to 32.0%. For both months, however, the most popular EDNS0 buffer value is 4096 bytes, with 40% of the resolvers using it.

Resolvers that adopted the DNS Flag Day value: We identified 6126 resolvers that changed their EDNS0 value to 1232 bytes, as can be seen in Table 4b. They belonged to 629 different ASes, but most of them (3422) belonged to only two ASes – one in Taiwan and the other in Poland.

Looking back to 1.5 years: The Flag Day 2020 was originally proposed in Oct. 2019. Given some operators may deploy it *before* the Flag Day chosen date (Oct. 1 2020), we analyze the proportion resolvers we see over more than 1.5 years (May 2019-Oct 2020). Figure 9 shows the percentage of unique IP addresses announcing different buffer sizes per day. From May 2019 to Oct. 2020, we see that despite the increase of resolvers using EDNS0 1232, they winding up accounting for only 4.4% of the total resolvers. 4096 byte resolvers reduced from 50% to 40%. These results show that a large population of resolvers still needs to be reconfigured to use EDNS0 1232 bytes.

5 Related Work

IP fragmentation: the problems related with IP fragmentation are well known [5]: it has problems with "middleboxes" (such as network address translation (NAT) devices, with stateless firewalls), by being expensive and error prone and may lead to unreachability [8,4,5,14]. It has also security vulnerabilities – it has been used DNS for cache poisoning attacks on DNS [16,45], and to compromise CAs based on it. Besides, there are several well-know attacks that exploit fragmentation [52,26,21,13]. Given these series of problems, IP fragmentation is considered fragile and should be avoided, also in DNS [5,12,53].

DNS and large responses: Large DNS/UDP responses have been previously shown to cause unreachability [49,46]. In 2011, using active measurements, Weaver *et al.* [49] have shown that 9% of clients could not receive fragmented DNS/UDP packets. Given our vantage point are not clients, we cannot determine this rate. We showed, however, the number of ICMP messages showing that DNS messages exceed the path MTU (§3.2). In a 2012 study [46], the authors analyzed DNSSEC messages (8.4M) from 230k resolvers to authoritative servers hosted SURFnet, the Dutch NREN. for 4k+ zones. They showed how 58% of resolvers received fragmented responses for DNSSEC queries.

Our results show a sharp contrast to both of these studies: by analyzing 114B queries from more than 3M resolvers, for one zone (`.nl`), we show a tiny fraction of fragmented queries (10k/day, §3.2). Besides, we also analyze truncation, responses sizes distribution, resolver behavior, EDN0 distribution, from two distinct large DNS anycast operators that provide DNS service to `.nl`. Another (non-academic) study from Google Public DNS operators in 2020 [24] showed similar rates of truncation and fragmentation to ours.

6 Conclusions

DNS/UDP large messages that lead to fragmentation have been long feared and blamed for causing unreachability. Drawing from 114B queries/responses, we asses state of affairs of large messages on DNS. We show that large responses are rare, and that server-side IP fragmentation is minimal (albeit most of fragmentation seems to take place still on the network). In case of clients experience query timeouts on DNS/UDP, we show that 75% of resolvers do fall back to TCP – and by this way are able to retrieve large responses. Previous research has shown that "hammering" and server switching – behaviors shown by resolvers in the wild – are expected to be useful in avoiding unreachability.

Still, our evaluation of more than 3M resolvers show that they still have a long way to go: many of them announce either small (512 bytes) or large (4096 bytes) EDNS0 buffer sizes, both leading to more truncation, and increasing the chances of fragmentation/packets being lost on the network.

We also show that the initial uptake of the DNS Flag Day 2020 suggested EDNS0 buffer size has not been very wide, however, similar to DNSSEC algorithms adoption, it would be interesting to evaluate this adoption over time, especially now that major resolver vendors have adopted this value.

References

1. 1.1.1.1: The Internet's Fastest, Privacy-First DNS Resolver. https://1.1.1.1/ (Apr 2018), https://1.1.1.1/
2. Abley, J., Gudmundsson, O., Majkowski, M., Hunt, E.: Providing Minimal-Sized Responses to DNS Queries That Have QTYPE=ANY. RFC 8482, IETF (Jan 2019), http://tools.ietf.org/rfc/rfc8482.txt
3. Arends, R., Austein, R., Larson, M., Massey, D., Rose, S.: DNS Security Introduction and Requirements. RFC 4033, IETF (Mar 2005), http://tools.ietf.org/rfc/rfc4033.txt
4. Bellis, R.: DNS Transport over TCP - Implementation Requirements. RFC 5966, IETF (Aug 2010), http://tools.ietf.org/rfc/rfc5966.txt
5. Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., Gont, F.: IP Fragmentation Considered Fragile. RFC 8900, IETF (Sep 2020), http://tools.ietf.org/rfc/rfc8900.txt
6. Brandt, M., Dai, T., Klein, A., Shulman, H., Waidner, M.: Domain Validation++ For MitM-Resilient PKI. p. 2060–2076. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243790
7. Damas, J., Graff, M., Vixie, P.: Extension Mechanisms for DNS (EDNS(0)). RFC 6891, IETF (Apr 2013), http://tools.ietf.org/rfc/rfc6891.txt
8. De Boer, M., Bosma, J.: Discovering Path MTU black holes on the Internet using RIPE Atlas. Master's thesis, University of Amsterdam (2012), https://nlnetlabs.nl/downloads/publications/pmtu-black-holes-msc-thesis.pdf
9. Deering, S., Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF (Dec 1998), http://tools.ietf.org/rfc/rfc2460.txt
10. Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., Wessels, D.: DNS Transport over TCP - Implementation Requirements. RFC 7766, IETF (Mar 2016), http://tools.ietf.org/rfc/rfc7766.txt
11. Elvy, M., Nedved, R.: Network mail path service. RFC 915, IETF (Dec 1984), http://tools.ietf.org/rfc/rfc0915.txt
12. Fujiwara, K., Vixie, P.: Serving Stale Data to Improve DNS Resiliency (*work in progress*). Internet Draft (Apr 2020), https://tools.ietf.org/html/draft-fujiwara-dnsop-avoid-fragmentation-03
13. Gont, F.: Security Implications of Predictable Fragment Identification Values. RFC 7739, IETF (Feb 2016), http://tools.ietf.org/rfc/rfc7739.txt
14. Gont, F., Linkova, J., Chown, T., Liu, W.: Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World. RFC 7872, IETF (Jun 2016), http://tools.ietf.org/rfc/rfc7872.txt
15. Google: Public DNS (Jan 2020), https://developers.google.com/speed/public-dns/
16. Herzberg, A., Shulman, H.: Fragmentation considered poisonous, or: One-domain-to-rule-them-all. In: 2013 IEEE Conference on Communications and Network Security (CNS). pp. 224–232. IEEE (2013)
17. Hoffman, P., Sullivan, A., Fujiwara, K.: DNS Terminology. RFC 8499, IETF (Jan 2019), http://tools.ietf.org/rfc/rfc8499.txt

18. Huston, G.: Dealing with IPv6 fragmentation in the DNS. https://blog.apnic.net/2017/08/22/dealing-ipv6-fragmentation-dns/ (Aug 2017)
19. Internet Assigned Numbers Authority (IANA): Root Files. https://www.iana.org/domains/root/files (2020)
20. ISC: 4. bind 9 configuration reference. https://bind9.readthedocs.io/en/v9_16_6/reference.html (2020)
21. Krishnan, S.: Handling of Overlapping IPv6 Fragments. RFC 5722, IETF (Dec 2009), http://tools.ietf.org/rfc/rfc5722.txt
22. Kulkarni, M., Patel, A., Leung, K.: Mobile IPv4 Dynamic Home Agent (HA) Assignment. RFC 4433, IETF (Mar 2006), http://tools.ietf.org/rfc/rfc4433.txt
23. Laurie, B., Sisson, G., Arends, R., Blacka, D.: DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. RFC 5155, IETF (Mar 2008), http://tools.ietf.org/rfc/rfc5155.txt
24. Lieuallen, A.: DNS Flag Day 2020 and Google Public DNS (Oct 2020), https://www.youtube.com/watch?v=CHprGFJv_WE
25. McPherson, D., Oran, D., Thaler, D., Osterweil, E.: Architectural Considerations of IP Anycast. RFC 7094, IETF (Jan 2014), http://tools.ietf.org/rfc/rfc7094.txt
26. Miller, I.: Protection Against a Variant of the Tiny Fragment Attack (RFC 1858). RFC 3128, IETF (Jun 2001), http://tools.ietf.org/rfc/rfc3128.txt
27. Mockapetris, P.: Domain names - concepts and facilities. RFC 1034, IETF (Nov 1987), http://tools.ietf.org/rfc/rfc1034.txt
28. Mockapetris, P.: Domain names - implementation and specification. RFC 1035, IETF (Nov 1987), http://tools.ietf.org/rfc/rfc1035.txt
29. Moura, G.C.M., Heidemann, J., Müller, M., de O. Schmidt, R., Davids, M.: When the dike breaks: Dissecting DNS defenses during DDoS. In: Proceedings of the ACM Internet Measurement Conference. pp. 8–21. Boston, MA, USA (Oct 2018). https://doi.org/https://doi.org/10.1145/3278532.3278534
30. Moura, G.C.M., Heidemann, J., de O. Schmidt, R., Hardaker, W.: Cache me if you can: Effects of DNS Time-to-Live. In: Proceedings of the ACM Internet Measurement Conference. pp. 101–115. ACM, Amsterdam, the Netherlands (Oct 2019). https://doi.org/https://doi.org/10.1145/3355369.3355568, https://www.isi.edu/%7ejohnh/PAPERS/Moura19b.html
31. Müller, M., Moura, G.C.M., de O. Schmidt, R., Heidemann, J.: Recursives in the wild: Engineering authoritative DNS servers. In: Proceedings of the ACM Internet Measurement Conference. pp. 489–495. ACM, London, UK (2017). https://doi.org/10.1145/3131365.3131366
32. OpenDNS: Setup Guide: OpenDNS. https://www.opendns.com/setupguide/ (Jan 2019), https://www.opendns.com/setupguide
33. Partridge, C., Mendez, T., Milliken, W.: Host Anycasting Service. RFC 1546, IETF (Nov 1993), http://tools.ietf.org/rfc/rfc1546.txt
34. Postel, J.: Internet Control Message Protocol. RFC 792, IETF (Sep 1981), http://tools.ietf.org/rfc/rfc0792.txt
35. Postel, J.: Internet Protocol. RFC 791, IETF (Sep 1981), http://tools.ietf.org/rfc/rfc0791.txt
36. Quad9: Quad9 — Internet Security & Privacy In a Few Easy Steps. https://quad9.net (Jan 2018)
37. Rekhter, Y., Li, T., Hares, S.: A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF (Jan 2006), http://tools.ietf.org/rfc/rfc4271.txt
38. RIPE NCC: RIPE Atlas measurement IDS. https://atlas.ripe.net/measurements/ID (Oct 2020), , where ID is the experiment ID: large:27759950, small:27760294

39. RIPE NCC: RIPE Atlas Probes. https://ftp.ripe.net/ripe/atlas/probes/archive/2020/05/ (May 2020)
40. RIPE NCC Staff: RIPE Atlas: A Global Internet Measurement Network. Internet Protocol Journal (IPJ) **18**(3), 2–26 (Sep 2015)
41. Root Server Operators: Root DNS (May 2020), http://root-servers.org/
42. SIDN Labs: ENTRADA - DNS Big Data Analytics (Jan 2020), https://entrada.sidnlabs.nl/
43. SIDN Labs: .nl stats and data (2020), https://stats.sidnlabs.nl/en/dnssec.html
44. Thomson, S., Huitema, C., Ksinant, V., Souissi, M.: DNS Extensions to Support IP Version 6. RFC 3596, IETF (Oct 2003), http://tools.ietf.org/rfc/rfc3596.txt
45. Tomas Hlavacek: IP fragmentation attack on DNS. In: RIPE 67, – Athens, Greece. https://ripe67.ripe.net/presentations/240-ipfragattack.pdf (October 2016)
46. Van Den Broek, G., Van Rijswijk-Deij, R., Sperotto, A., Pras, A.: DNSSEC meets real world: dealing with unreachability caused by fragmentation. IEEE Communications Magazine **52**(4), 154–160 (2014)
47. Van Den Broek, G., van Rijswijk-Deij, R., Sperotto, A., Pras, A.: DNSSEC meets real world: dealing with unreachability caused by fragmentation. IEEE communications magazine **52**(4), 154–160 (2014)
48. Vixie, P.: Extension Mechanisms for DNS (EDNS0). RFC 2671, IETF (Aug 1999), http://tools.ietf.org/rfc/rfc2671.txt
49. Weaver, N., Kreibich, C., Nechaev, B., Paxson, V.: Implications of Netalyzr's DNS measurements. In: Proceedings of the First Workshop on Securing and Trusting Internet Names (SATIN), Teddington, United Kingdom. Citeseer (2011)
50. Wijngaards, W.: release-1.12.0: Unbound 1.12.0. https://github.com/NLnetLabs/unbound/releases/tag/release-1.12.0 (2020)
51. Wullink, M., Moura, G.C., Müller, M., Hesselman, C.: Entrada: A high-performance network traffic data streaming warehouse. In: Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP. pp. 913–918. IEEE (Apr 2016)
52. Ziemba, G., Reed, D., Traina, P.: Security Considerations for IP Fragment Filtering. RFC 1858, IETF (Oct 1995), http://tools.ietf.org/rfc/rfc1858.txt
53. Špaček, P., Surý, O.: DNS flag day 2020. https://dnsflagday.net/2020/ (October 2020)

A Extra graphs

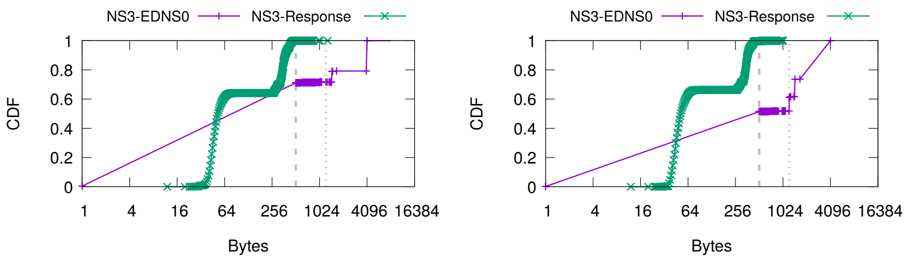
Figure 10 shows the truncated queries for NS3 in 2020. Figure 11 shows the timeseries of truncated queries for `.nl` on July 2019.

We see in the same figures a close match between UDP truncated queries and TCP ones – however not quite the same.

Figure 11 shows the CDF of DNS/UDP truncated queries for 2019, per server.

B Clients and large DNS/UDP responses

We evaluate if DNS messages are being lost along the way from authoritative servers to clients. To do that, we setup two measurements using RIpe Atlas (∼10k probes), as shown in Table 5. We configure each probe to send a query directly



(a) NS3: IPv4 (b) NS3: IPv6

Fig. 10: NS3: CDF of DNS/UDP TC responses for `.nl`: July 2020

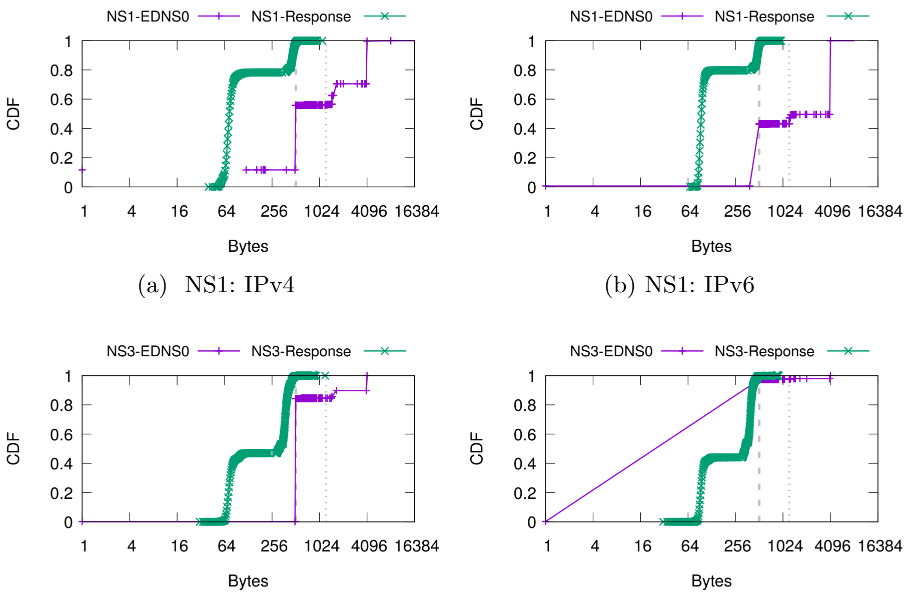
to NS3, the server that returns additional records. As such, probes *bypass* local resolvers, so they cannot fallback to TCP: they simply send one UDP query. We setup two measurements: one that retrieves *large* DNS/UDP responses (1744 bytes, Large column) and one that retrieves *small* ones (221 bytes).

	Large	Small
EDNS0 buffer	4096	512
Query	ANY NS `.nl`	A ns1.dns.`nl`
Target	ns3.dns.`nl`	
Response Size	1744	221
Protocol/IP	UDP/IPv4	
Active Probes	9323	9322
∩	8576	
Queries	557047	555007
∩	512351	510575
OK	473606	497792
timeout	38745(6.9%)	12783 (2.5%)

Table 5: Atlas measurements for large and small responses. Datasets:[38]

In total, we see 8576 probes being active on both measurements – sending more than 1M queries (512k on the Large, 510k on the Small). For each probe, we look then into the number of failed responses (timeout), for the small and large measurements. We see that 6.9% of queries timeout for the large measurement, however, 2.5% of them also timeout for short responses.

Next we investigate each probe. e compute the percentage of timeout queries per dataset. We then compute the difference between the rate of failed queries for the large and the small datasets. Figure 12 shows the results. Out of the 8576 probes on both datasets, 6191 have no error difference for both large and small queries (72%). 10% in fact have more errors for the small dataset query,



(a) NS1: IPv4 (b) NS1: IPv6

(c) NS3: IPv4 (d) NS3: IPv6

Fig. 11: CDF of DNS/UDP TC answers for `.nl`: July 2019

and only 17% have more errors for the longer answers. 325 have 100% of errors for the large datasets, but no errors for the small datasets.

Overall, this measurement show the fragmentation is still an issue on the client side –which justifies the flag day.

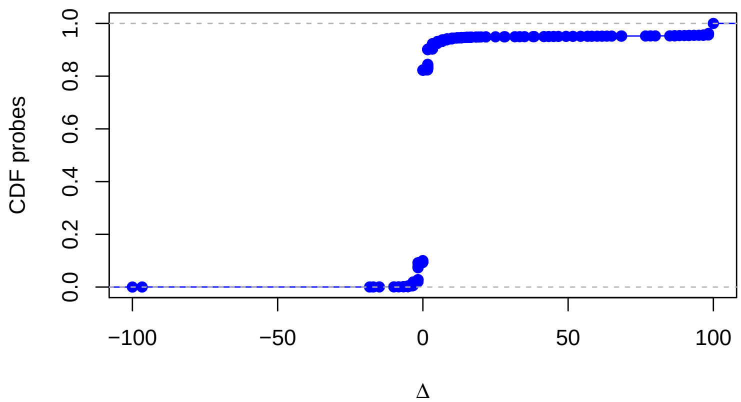


Fig. 12: Error CDF per Atlas Probe, for Large and Small response datasets.