

Measuring DNS over TCP in the Era of Increasing DNS Response Sizes: A View from the Edge

Mike Kosek

Technical University of Munich
kosek@in.tum.de

Simon Huber

Technical University of Munich
simon.huber@tum.de

Trinh Viet Doan

Technical University of Munich
doan@in.tum.de

Vaibhav Bajpai

CISPA Helmholtz Center for Information Security
bajpai@cispa.de

ABSTRACT

The Domain Name System (DNS) is one of the most crucial parts of the Internet. Although the original standard defined the usage of DNS over UDP (DoUDP) as well as DNS over TCP (DoTCP), UDP has become the predominant protocol used in the DNS. With the introduction of new Resource Records (RRs), the sizes of DNS responses have increased considerably. Since this can lead to *truncation* or *IP fragmentation*, the fallback to DoTCP as required by the standard ensures successful DNS responses by overcoming the size limitations of DoUDP. However, the effects of the usage of DoTCP by stub resolvers are not extensively studied to this date. We close this gap by presenting a view at DoTCP from the Edge, issuing 12.1M DNS requests from 2,500 probes toward *Public* as well as *Probe* DNS recursive resolvers. In our measurement study, we observe that DoTCP is generally slower than DoUDP, where the relative increase in *Response Time* is less than 37% for most resolvers. While optimizations to DoTCP can be leveraged to further reduce the response times, we show that support on *Public* resolvers is still missing, hence leaving room for optimizations in the future. Moreover, we also find that *Public* resolvers generally have comparable reliability for DoTCP and DoUDP. However, *Probe* resolvers show a significantly different behavior: DoTCP queries targeting *Probe* resolvers fail in 3 out of 4 cases, and, therefore, do not comply with the standard. This problem will only aggravate in the future: As DNS response sizes will continue to grow, the need for DoTCP will solidify.

CCS CONCEPTS

• **Networks** → **Network measurement**; *Transport protocols*;

KEYWORDS

Domain Name System, Transmission Control Protocol, DNS over TCP, RIPE Atlas

1 INTRODUCTION

The Domain Name System (DNS) is one of the most crucial parts of the Internet, taking part in almost every connection of any service. The original standard defined the usage of DNS over UDP (DoUDP) as well as DNS over TCP (DoTCP) [28, 29]. However, UDP has become the predominant protocol used in the DNS [38, 39] due to its latency benefits, given its absence of connection establishment and state handling.

With the introduction of new RRs such as AAAA (IPv6 support) [34] or RRSIG (DNSSEC) [49], the sizes of DNS responses have increased considerably [27, 35]. The most recent efforts to establish a generic format within the DNS to provide clients with information on how to access a service using the SVCB (Service Binding) RR [51], which also provides the configuration required for *TLS Encrypted Client Hello* [45], will continue this trend of increasing DNS response sizes.

To increase the original DoUDP response size limit of 512 bytes [29], the *Extension Mechanisms for DNS (EDNS(0))* [14, 56] were introduced to allow requests and responses of up to 65,535 bytes. However, when a DoUDP request or response exceeds the limit of either the original 512 bytes or the *EDNS(0)* size signaled, it is marked as *truncated*, which results in fallback to DoTCP [6, 17]: Due to the connection-oriented nature of TCP, DoTCP overcomes the size limitations of DoUDP and ensures a successful DNS response.

Several studies investigate the *EDNS(0) Buffer Sizes* used by requests issued from recursive resolvers to authoritative servers, finding that buffer sizes falling short of or exceeding the recommended limits remain the predominant sizes [38, 39]. While this poses a risk for *truncation* as well as *IP fragmentation*, the effects of these issues on the DNS are extensively studied [7, 22, 39, 55, 57]. Inherently, DNS over TLS (DoT) and DNS over HTTPS (DoH) circumvent these issues by using TCP as the underlying transport protocol. However, their adoption by recursive resolvers is still low [16, 36, 52], and both protocols aid the trend of Internet centralization [19, 38]. Hence, the need for DoTCP will solidify in the future. A contemporary study on DoTCP [37] has looked at its support in the wild; the authors find a lack of proper TCP fallback and DoTCP adoption in numerous cases, although resolvers should already support DoTCP as required by the standard [17]. In general, the effects of DoTCP usage by stub resolvers in terms of *Failure Rates* and *Response Times* are not yet extensively studied.

We close this gap by presenting a unique view on DNS over TCP from the Edge, evaluating *Failure Rates* (see § 4) and *Response Times* (see § 5). Using the RIPE Atlas (RA) platform [46], we issue 12.1M DNS requests from the stub resolvers of 2,500 probes toward *Public* as well as *Probe* DNS recursive resolvers over both DoTCP and DoUDP (see § 3).

Failure Rates. While failure rates over DoTCP are comparable with DoUDP for *Public* resolvers, DoTCP failure rates for *Probe* resolvers are significantly higher. As such, DoTCP queries targeting *Probe* resolvers fail in 3 out of 4 cases, and, therefore, do not comply with the standard.

With respect to the largest Autonomous Systems (ASes) in terms of probes, we find that failure rates over DoTCP for most pairings of ASes and *Public* resolvers are low, roughly matching the respective failure rates over DoUDP. However, our observation also hints at path-specific issues between the COMCAST and ORANGE ASes and OpenNIC, where nearly all DoTCP AND DoUDP requests fail. Looking at *Probe* resolvers, we again observe high failure rates over DoTCP across all ASes, indicating that *Probe* resolvers still lack reliable and vast support for DoTCP.

Response Times. Response times over DoTCP are highly varying depending on the continent of the RA probes location for *Public* and *Probe* resolvers. Overall, we find DoTCP to be slower than DoUDP for nearly all pairings of continent and resolver. However, when considering the response time differences between both protocols, the relative increase over all continents is less than 37% for most *Public* and *Probe* resolvers.

Moreover, our evaluation shows that *Public* resolver lack DoTCP optimization, not offering support for *EDNS0 TCP keepalive* and *TCP Fast Open (TFO)*, with the latter only being supported by Google. However, using the TFO cookie in subsequent connections to Google was only successful in rare cases: Due to the connection reset following the refused TFO cookie, the usage of TFO on Google actually increases the response time in the majority of cases.

Outline. In § 2 we present related work, followed by our methodology and an overview of our dataset in § 3. We discuss our failure rate analysis in § 4, followed by the response time analysis in § 5. Limitations and future work are discussed in § 6 before we conclude the paper with § 7.

2 RELATED WORK

Extension Mechanisms for DNS (EDNS(0)) are now commonly used in the DNS [8, 32], e.g., to add more information to a DNS message, thereby increasing its size. However, DNS requests and responses exceeding the path Maximum Transmission Unit (MTU) cause *IP fragmentation*, which can lead to unreachability due to firewalls blocking fragmented IP packets, or failures due to recipients being unable to reassemble them [5, 17, 55]. In addition to the operational challenges this poses, several studies have shown that cache poisoning attacks using *IP fragmentation* can modify DNS responses [7, 22].

To avoid *IP fragmentation*, multiple proposals exist to restrict the DoUDP payload size through *EDNS(0) Buffer Size* values. These limits should be in ranges of 1,220–1,472 bytes for IPv4 and 1,220–1,452 bytes for IPv6 [20, 26, 30]. Most notably, the *DNS flag day 2020* [18] proposed a limit of 1,232 bytes for both IPv4 and IPv6, based on the minimum IPv6 MTU of 1,280 bytes.

Recent studies have shown that *EDNS(0) Buffer Sizes* of 1,232 bytes or less are already widely used in requests from recursive resolvers to authoritative servers, although 512 and 4,096 bytes remain the predominant sizes [38, 39]. While buffer sizes of 512 bytes pose the risk to trigger a *truncation* early, 4,096 bytes exceed the predominant path MTU of 1,500 bytes in the Internet [17] and, therefore, pose a risk for *IP fragmentation*. With DNS response sizes becoming larger in general [27, 35], *truncation* as well as *IP fragmentation* rates will likely increase, and ultimately lead to increased DoTCP usage in the future.

Several studies investigate the effects of *truncation* [39, 55, 57] and *IP fragmentation* [7, 20, 22, 39] on DNS, although they do not focus on DoTCP in detail. A general recommendation is to prefer DoTCP over DoUDP in order to avoid *IP fragmentation* in the DNS [5, 18, 20]. However, a recent study [15] shows that ICMP messages can be leveraged to trigger *IP fragmentation* on DoTCP as well, thereby questioning this recommendation.

Other related work studies the adoption and performance of novel DNS protocols like DoT [16, 25, 36, 52] as well as DoH [16, 25, 36], and how they compare to DoUDP [25, 52] in terms of *Failure Rates* and *Response Times*. However, researchers as well as operators express concerns due to both protocols aiding the trend of Internet centralization [19, 38]. Nevertheless, both DoT and DoH inherently solve the issues of *truncation* and *IP fragmentation* in the DNS, yet, their adoption by recursive resolvers is still low [16, 36, 52].

Considering these evolutions in the DNS, DoTCP is needed to prevent *truncation* and *IP fragmentation* until DoT or DoH are more widely adopted. While DoTCP should already be usable to date as required by the standard [17], the effects of its usage in terms of *Failure Rates* and *Response Times* are not yet extensively studied.

3 METHODOLOGY AND DATASET

To study DoTCP from the Edge, we perform distributed DNS measurements using the RIPE Atlas (RA) platform [46].

3.1 Measurement Design

Measurement Probes. We select RA hardware probes with the *home* user tag excluding *Anchor* probes: The *home* tag is used to identify RA probes operating in residential home networks, of which we find 3,364 probes. Additionally, we only select probes using hardware version 3 or later due to possible link issues [2, 24]. Of the remaining 2,815 probes, we select 2,500 probes randomly for our measurement study. From these targeted 2,500 probes, 2,363 probes ultimately execute the measurements (of which 2,361 include location information); the remaining probes were offline or were not considered by the RA scheduler during the measurement period. The final set of probes is distributed across the globe in 83 different countries and 655 distinct ASes. Table 1 shows the absolute and relative number of the probes per continent and per AS for the top 10 ASes based on the number of probes. The relative number is calculated based on 2,361 probes with location information for the continent-based analysis, whereas the AS-based analysis covers 2,363 probes in total. Note that the locations of the probes are biased toward Europe (EU) and North America (NA), as most of the used probes (88.01%) are deployed in these continents. Similarly, the top 10 source ASes connect more than a quarter of the used probes (27.04%) and are also biased towards EU and NA: While COMCAST, ATT, and UUNET are NA-based, the remaining 7 ASes are EU-based. Hence, our observations are limited by the probes' locations and networks (see § 6).

Public Resolvers. For the measurement targets, we select 10 *Public* recursive resolvers based on their usage in related work (see § 2), querying the IPv4 anycast addresses listed in Table 2.

Probe Resolvers. In addition to the public recursive resolvers, we also query the recursive resolvers configured locally on the probes for comparison. While every RA probe can be configured

Table 1: Distribution of 2,361 RIPE Atlas probes by geographical location (continent) of the probes (top) and by AS for the top 10 ASes based on number of probes (bottom).

Type	Location	Number of Probes
Continent	Europe (EU)	1,636 (69.29%)
	North America (NA)	442 (18.72%)
	Asia (AS)	149 (6.31%)
	Oceania (OC)	71 (3.01%)
	South America (SA)	33 (1.40%)
	Africa (AF)	30 (1.27%)
Autonomous System	DTAG (AS3320)	127 (5.37%)
	COMCAST (AS7922)	96 (4.06%)
	VODANET (AS3209)	93 (3.94%)
	PROXAD (AS12322)	89 (3.77%)
	ORANGE (AS3215)	70 (2.96%)
	ATT (AS7018)	39 (1.65%)
	UUNET (AS701)	35 (1.48%)
	TNF (AS33915)	33 (1.40%)
	NTL (AS5089)	29 (1.23%)
	IBSNAZ (AS3269)	28 (1.18%)
others	1,724 (72.96%)	

Table 2: Evaluated Public Recursive Resolvers and their queried IPv4 Anycast Addresses

Public Recursive Resolver	IPv4 Anycast Address
CleanBrowsing	185.228.168.9
Cloudflare DNS	1.1.1.1
Comodo Secure DNS	8.26.56.26
Google Public DNS	8.8.8.8
Neustar UltraDNS	64.6.64.6
OpenDNS	208.67.222.222
OpenNIC	185.121.177.177
Quad9	9.9.9.9
UncensoredDNS	91.239.100.100
Yandex.DNS	77.88.8.8

with multiple resolvers, every DNS measurement which sets the `use_probe_resolver` option is issued to every locally configured resolver. Hence, one DNS measurement request might result in more than one result per probe; the average number of configured resolvers per probe was 2.1. Based on the source IP address of the DNS responses, we exclude any known unicast or anycast IP address used by the *Public* resolvers listed in Table 2 for a more accurate distinction and comparison: The final set is denoted as *Probe* resolvers and represents Internet Service Provider (ISP) as well as alternative public DNS services.

DNS Queries. We issue DNS queries for A records using DoTCP as well as DoUDP. For these queries, we include the *EDNS(0)* OPT RR with an *EDNS(0) Buffer Size* of 1,232 bytes as proposed by the *DNS flag day 2020* [18], thereby signaling support to receive DNS responses of up to 1,232 bytes.

To cover a diversity of different domains, which allows us to investigate possible differences between their popularity and region, we construct a set of 200 domains from the Alexa Global and Country Top lists as of March 29, 2021 [1]. We sample 150 popularity-focused domains by splitting the Global Top 1M list into 10 evenly-sized bins of 100k each (by rank order) and select the 15 highest-ranked domains of each bin. For the remaining 50 domains, we determine the countries with the highest numbers of deployed RA probes, additionally considering all continents. For each of those countries (BR, DE, GB, IT, JP, NL, NZ, RU, US, ZA), we choose the 5 highest-ranked domains of the associated country code Top-Level Domain (e.g., `.br`, `.de`, `.co.uk`), ultimately resulting in 50 region-focused domains. Nevertheless, as we do not find any significant deviations for *Failure Rates* or *Response Times*, we do not further distinguish between the domains in our analyses.

To counter limitations of a) the RA platform, which enforces the *Recursion Desired* bit on measurements to *Probe* but not to *Public Resolvers* [47], as well as b) *Public* resolvers utilizing different caching strategies that cannot ensure cached records when using the RA platform [44], we issue queries to be resolved recursively in order to ensure uncached DNS responses, which enables comparable results of *Public* and *Probe* resolvers. This is achieved by two means:

- (1) *Unique Prefixes.* We add *Unique Prefixes* to our 200 domains, which consist of the probe ID and the timestamp of the DNS request.
- (2) *Recursion Desired.* For *Probe Resolver* measurements, the *Recursion Desired* (RD) bit is set by default and enforced on RIPE Atlas for privacy protection [47]. However, the bit is *NOT* set by default for *Public* resolver measurements, so we explicitly set the RD bit for all measurements.

While (1) ensures that the queried domain does not exist and is, therefore, not cached, (2) ensures that the resolver will recursively resolve the requested domain. If *Recursion Desired* would *NOT* be set, a query would *NOT* be recursively resolved but instead be directly responded to by the resolver, even if the queried domain was *NOT* cached or a wildcard matched the queried domain [23, 44]. Hence, setting *Recursion Desired* on all measurements is required to compare *Public* to *Probe* resolvers. Moreover, as the overhead of the authoritative resolver lookup is identical on both DoTCP and DoUDP, the overhead is canceled out for both protocols when analyzing the *differences* in *Response Times* as presented in § 5, which enables the comparison of DoTCP and DoUDP measurements.

Ethical Considerations. The measurement study does not raise any ethical concerns, as we exclusively use RIPE Atlas probes hosted by volunteers, who explicitly agree with publication of the collected data through the RIPE Atlas Service Terms and Conditions [48]. We further query unique domains and set the *RD* bit to ensure recursive name resolution and to avoid *cache snooping*. Moreover, we aggregate the collected data for the analyses and do not discuss individual probes (or their IP addresses or location coordinates) to preserve the privacy of the voluntary probe hosts.

Reproducibility. In order to enable the reproduction of our findings [3], we make the raw data of our measurements as well as the analysis scripts and supplementary files publicly available on GitHub¹. Please also refer to the appendix for detailed instructions.

¹<https://github.com/kosekmi/2022-ccr-dns-over-tcp-from-the-edge>

Table 3: Dataset Overview: Sample Sizes, Failure Rates, Failure Reasons, and EDNS(0) Buffer Sizes for Public and Probe recursive resolvers for DoTCP and DoUDP.

	Public DoTCP	Public DoUDP	Probe DoTCP	Probe DoUDP
Samples				
– Total	4,655,635	4,656,086	454,151	454,417
– Successful	4,282,559	4,279,568	113,728	447,009
– Failure Rate	8.01%	8.09%	74.96%	1.63%
Failure Reasons				
– TUNCONNECT	4.79%	-	74.72%	-
– Timeout	3.22%	8.09%	0.24%	1.63%
– Socket	-	>0.01%	-	-
– other	>0.01%	-	-	-
EDNS(0) Buffer Sizes in bytes				
– 512	26.52%	26.35%	42.11%	28.43%
– 1232	37.51%	44.73%	31.60%	29.93%
– 4096	35.59%	28.22%	21.61%	27.61%
– other	0.29%	0.31%	3.34%	8.17%
– none	0.09%	0.39%	1.33%	5.87%

3.2 Dataset Overview

Dataset Preparation. Overall, we issue a total of 12.1M DNS queries (2,500 probes \times (10 Public Resolvers + on avg. 2.1 Probe Resolvers) \times 200 domains (with 1 query per domain) \times 2 Protocols with DoUDP and DoTCP) as part of our measurement study in April 2021. As stated in § 3.1, 2,363 probes execute the measurements and remain in the analysis dataset. While we explicitly state the IPv4 addresses to be used by our requests to Public resolvers, recall that requests to Probe resolvers are issued to every locally configured resolver, hence, also over IPv6. As we focus on IPv4 exclusively in this paper, we leave a comparative study between IPv4 and IPv6 open for future work. Thus, we exclude all measurements with IPv6 destination addresses (17,556 samples).

In total, we take 4,655,635 Public DoTCP, 4,656,086 Public DoUDP, 454,151 Probe DoTCP, and 454,417 Probe DoUDP samples into account for our analyses (see Table 3).

EDNS(0) Buffer Sizes. The EDNS(0) Buffer Size option allows a DoUDP packet to extend its size beyond the default 512 bytes [29], where the signaled buffer size should represent the maximum UDP payload size which the network of the sender can handle [14, 56]. For our queries, we include an EDNS(0) Buffer Size (see § 3.1) in order to check whether the Public and Probe recursive resolvers support extended buffer sizes through EDNS(0). If supported, the recursive resolver signals its maximum EDNS(0) Buffer Size back to the requestor, i.e., the maximum UDP payload size which the resolver’s network stack should be able to process. While the resolver knows both the maximum EDNS(0) Buffer Size of the requestor as well as its own, the resolver should use the minimum of both signaled values for the actual DNS response so that both endpoints can process the packets accordingly. Nevertheless, as the signaled EDNS(0) Buffer Sizes only represent the maximum buffer sizes that the endpoints should support, the actual size of the response can still exceed the path MTU. Moreover, the EDNS(0) Buffer Size is

often configured manually [4, 53], defaulting to sizes which might not be supported by the network in the first place.

Recent work (see § 2) studies the usage of EDNS(0) Buffer Sizes for DNS requests issued from recursive resolvers to authoritative servers [38, 39]. In these studies, the authors also observe the rate of DoTCP usage on their authoritative servers vantage points: In the first study [38], the authors focus on DNS cloud providers and find that DoTCP is used in up to 15% of requests issued by Facebook. In comparison, other evaluated providers (Amazon, Cloudflare, Google, Microsoft) show a usage of only 5% or below as of 2020. Similarly, the second study [39] shows a DoTCP usage of around 3–5% of requests as of 2020. In addition, the authors also evaluate constructed responses of authoritative servers to stub resolvers with a DoUDP size of 1,744 bytes, finding that 6.9% of responses and 3.9% of probes timed out, and, thus, lead to DoTCP fallback.

While we do not control the actual size of the DNS responses, we are not able to quantify the actual occurrence of DoTCP fallback on responses from recursive to stub resolvers (see § 6). However, our observations complement the aforementioned related studies (between recursive resolvers and authoritative servers) by presenting the signaled EDNS(0) Buffer Sizes for DoUDP requests issued from recursive to stub resolvers, for which we observe a similar distribution. Since the buffer sizes stated in Table 3 are only relevant for DoUDP, we did not analyze the observed differences between DoUDP and DoTCP. Hence, we detail our observations using DoUDP in the following, but include DoTCP for completeness.

We measure that 44.73% of DoUDP requests issued to Public resolvers and 29.93% of DoUDP requests issued to Probe resolvers respond with a buffer size of 1,232 bytes, which complies with the suggested value of the DNS flag day 2020 [18] and also honors the limits discussed by other proposals [20, 26, 30]. Notably, unbound [53] as well as BIND9 [4] changed their default EDNS(0) Buffer Sizes to 1,232 bytes following the DNS flag day in 2020. However, 26.35% of Public and 28.43% of Probe resolvers respond with a buffer size of 512 bytes, and 28.22% of Public and 27.61% of Probe resolvers respond with 4,096 bytes.

Most Public resolvers use a single EDNS(0) Buffer Size predominantly (>95%). Cloudflare, UncensoredDNS, and Yandex primarily use 1,232 bytes, while Comodo and OpenDNS use 4,096 bytes. On the other hand, CleanBrowsing and Google mainly use 512 bytes, whereas OpenNIC (75.4% with 4,096 bytes; 23.8% with 1,232 bytes), Quad9 (48.5% with 1,232 bytes; 49.8% with 512 bytes), and Neustar (50.2% with 1,232 bytes; 49.0% with 4096 bytes) show a mixed usage of buffer sizes instead.

Notably, our observations on Google responding with an EDNS(0) Buffer Size of 512 bytes in 98.0% of cases does differ to the observation made by [38]. The authors find that 24% of requests from Google to authoritative servers use EDNS(0) Buffer Sizes of up to 1,232 bytes, with the remaining 76% primarily using 4,096 bytes instead. Mapping our results to these observations shows that Google uses different EDNS(0) Buffer Sizes on stub-facing resolvers in comparison to authoritative-facing resolvers.

Other EDNS(0) Buffer Sizes are seen in 0.31% of cases for Public resolvers, and 8.17% of cases for Probe resolvers. Public resolvers show no buffer size greater than 4,096 bytes; in contrast, Probe resolvers exceed this value in 0.95% of cases with buffer sizes of 8,192 bytes (0.34%), 65,494 bytes (0.58%), and 65,535 bytes (0.03%).

Our observations show, that DNS responses from recursive to stub resolvers use *EDNS(0) Buffer Sizes* of 512 and 4,096 bytes in more than 55% of cases, thereby falling considerably short of or exceeding the recommended limits. While these results allow us to put our observations into perspective, a comprehensive study on *truncation* and *IP fragmentation* for requests issued from stub to recursive resolvers is left for future work (see § 6).

4 FAILURE RATES

In order to assess the reliability of DoTCP, we study the number of failures which the probes observe during their measurements. We define a measurement as failed if the probe did NOT receive a response from the queried resolver, and state the failure reasons according to the data provided by RIPE Atlas (RA): either due to issues with the TCP connection (*TUCONNECT*), with receiving a DNS response within 5 seconds (*Timeout*), or with sending the DNS request (*Socket*). We then determine the *Failure Rate* and *Failure Reasons* as the relative number of failures based on all measurements. Table 3 lists the overall *Failure Rates* and *Failure Reasons* by *Public* and *Probe* resolver measurements for both DoTCP and DoUDP.

Public resolvers exhibit similar *Failure Rates* for both DoTCP (8.01%) and DoUDP (8.09%), showing that the reliability of DoTCP is comparable to that of DoUDP. In terms of *Failure Reasons*, almost all failures of DoUDP measurements to *Public* resolvers are attributed to *Timeout*, whereas DoTCP measurements to *Public* resolvers show *TUCONNECT* as the primary failure reason with 4.79%.

Note that RA does not provide more detail on *TUCONNECT* errors; previous work [52] on DoT measurements using RA suggested that these failures are related to TLS negotiation errors. However, since we observe this behavior using TCP as well, it is more likely that *TUCONNECT* hints at issues with the (underlying) TCP connection instead, i.e., the RA probe is not able to establish a TCP connection with the recursive resolver (which in return causes a potential TLS negotiation to also fail).

More specifically, we attribute *TUCONNECT* to instances where the probe is informed about the unreachability of the contacted IP:Port combination by receiving a TCP RST to the probes TCP SYN packet, hence stating that the recursive resolver does not support DoTCP. On the other hand, a *Timeout* is recorded if loss occurs, or if no TCP RST is received, either due to not being elicited in the first place or due to being lost in transit. To substantiate this hypothesis, we issue RA DoTCP requests targeting a controlled recursive resolver: Thus, we are able to verify that elicited TCP RST packets result in *TUCONNECT*, whereas RA reports *Timeout* if no packet was sent in response to the TCP SYN. Please note that both failures might also occur if middleboxes drop the request, either silently (resulting in *Timeout*), or by eliciting a TCP RST packet themselves (resulting in *TUCONNECT*). Since we measure DoTCP from the edge, we cannot analyze possible path influences in more detail (see § 6).

Evaluating *Probe* resolvers, a significantly different behavior is shown in comparison to *Public*: For DoUDP, *Probe* resolvers have a fairly low failure rate with 1.63%. However, measurements attempting DoTCP fail in 74.96% of the cases, with *TUCONNECT* accounting for 74.72% (remaining 0.24% *Timeout*), which indicates that vast support for DoTCP among *Probe* resolvers is lacking.

In particular, RFC 7766 [17] states that implementations of authoritative servers, recursive resolvers, and stub resolvers MUST support DoTCP. In addition to the dominance of the *TUCONNECT* failure reason, the high failure rate of *Probe* resolvers for DoTCP in contrast to *Public* resolvers also indicates that these failures occur due to missing DoTCP support on the side of *Probe* resolvers. We suspect that this is due to most *Probes* using Customer-premises equipment (CPE) devices (e.g., home routers) as their resolvers, which typically forward DNS queries to an upstream DNS service operated by the ISP [50]: Out of the 74.96% DoTCP measurements that failed for the *Probe* resolvers (see Table 3), 99.47% were issued to resolvers with private IPv4 addresses (i.e., 74.56% of all DoTCP measurements to *Probe* resolvers). Thus, the observation indicates that almost all of the measurements to *Probe* resolvers are forwarded from CPE devices to ISP resolvers which do not implement DoTCP, and therefore do not comply with the standard defined by RFC 7766 [17].

Takeaway: While we find failure rates over DoTCP to be comparable with DoUDP for *Public* resolvers, DoTCP failure rates for *Probe* resolvers are significantly higher. As such, *Probe* resolvers cannot successfully return large DNS responses that require a fallback to DoTCP in 3 out of 4 cases, and, thus, do not comply with the standard.

By Continent. In order to investigate regional differences for DoTCP, we group the failures for each resolver and continent (based on geographic coordinates pulled from the RA probe API), and calculate the respective *Failure Rates*, as shown in Fig. 1 (top). While the top row aggregates all *Public* resolvers, each of the 10 *Public* resolvers as well as the *Probe* resolvers are detailed in the remaining rows. Note that this layout is the same for all remaining (sub)plots in the paper.

We find that DoTCP *Failure Rates* vary between different resolvers, as the total *Failure Rates* across all continents (top left) are within the range of 1.2–2.8% for CleanBrowsing, Cloudflare, Google, OpenDNS, Quad9, and Yandex. On the other hand, Comodo, Neustar, OpenNIC, and UncensoredDNS show considerably higher *Failure Rates* with 10.8–23.3%. Notably, Comodo and UncensoredDNS also show *TUCONNECT* as the primary *Failure Reason* with 95% and 92% of failures, respectively. In contrast, we observe mixed *Failure Reasons* with comparable occurrences of *TUCONNECT* and *Timeout* for the remaining *Public* resolvers. This indicates that DoTCP is not offered universally on all Points of Presence (PoPs) of the *Public* resolvers, as our observations show no clear preference for one specific failure reason.

Further, *Failure Rates* for a specific resolver also differ between continents (Fig. 1 top middle): E.g., we observe higher *Failure Rates* for most resolvers in South America (SA) and Asia (AS). *Probe* resolvers have *Failure Rates* of 63.7–78.1% across all continents, with resolvers in SA showing the lowest *Failure Rates*. We also find outliers in EU and NA, where Comodo (EU: 31.5%) and OpenNIC (NA: 45.3%) have significantly higher *Failure Rates* for DoTCP; OpenNIC further exhibits a higher failure rate in SA with 69.7%.

Comparing DoTCP and DoUDP, Fig. 1 (bottom) presents the absolute *Failure Rate* differences (in percentage points) between the DoTCP and DoUDP measurements for the resolvers and continents (bottom middle); i.e., positive values (colored in red) indicate higher *Failure Rates* for DoTCP, whereas negative values (colored

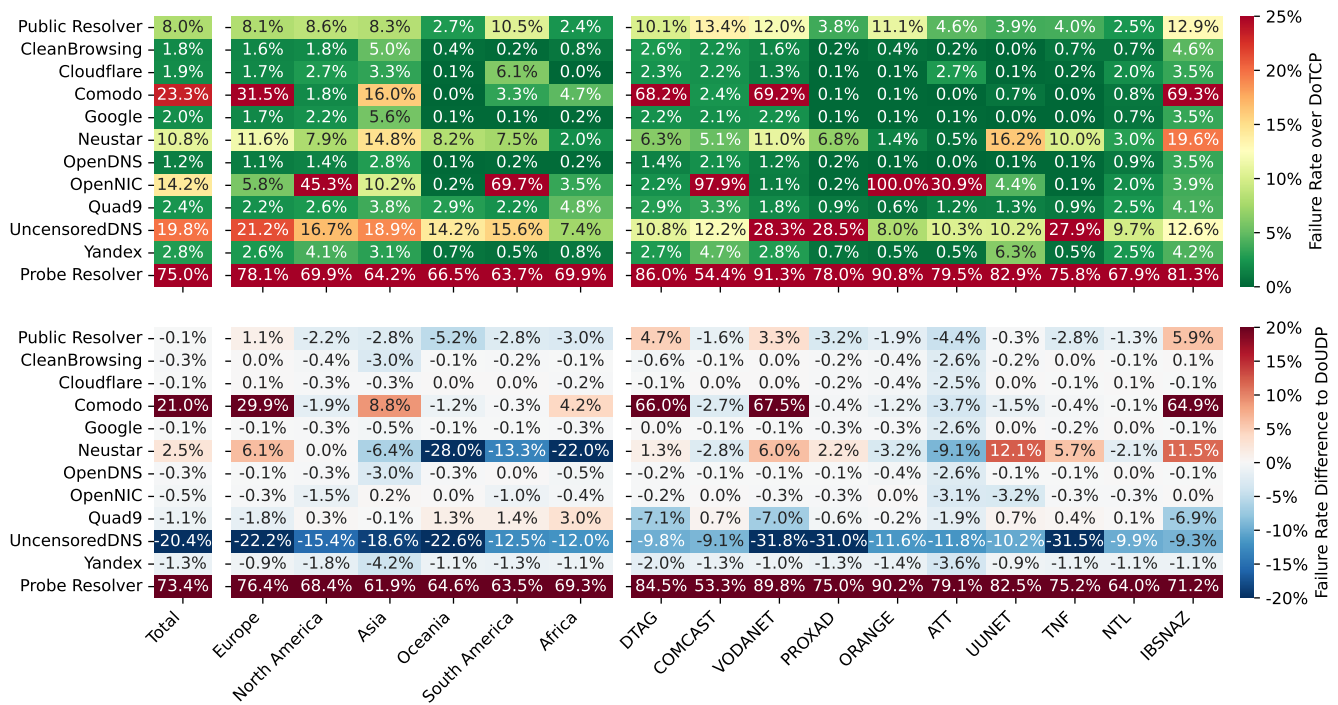


Figure 1: Failure rate by Resolver over DoTCP (top), along with respective failure rate difference in percentage points between DoTCP and DoUDP (bottom); across all samples in total (left), by Continent (middle), and by Top 10 ASes (right), each in descending order by number of probes. Positive values (colored in red) indicate higher failure rates for DoTCP.

in blue) represent higher *Failure Rates* over DoUDP instead. Overall, the differences between DoUDP and DoTCP are marginal for the resolvers showing low DoTCP failure rates, with the differences being around zero percentage points. However, probes across all continents (bottom left) experience lower failure rates to UncensoredDNS for DoTCP instead of DoUDP, as the DoUDP *Failure Rates* are higher by 12.0–22.6 percentage points in comparison. Similarly, measurements to Neustar failed more frequently over DoUDP (by 6.4–28.0 percentage points) for all continents except for EU and NA. On the other hand, Comodo exhibits much higher *Failure Rates* over DoTCP than over DoUDP for AS (8.8 percentage points) and EU (29.9 percentage points), which results in an overall difference of 21.0 percentage points in favor of DoUDP across all continents for Comodo. For *Probe* resolvers, the DoTCP *Failure Rates* are also significantly higher: The *Failure Rate* differences range from 61.9 to up to 76.4 percentage points; considering the absolute results described above (see Fig. 1 top middle), which barely differ from the percentages shown in the subtraction plot (see Fig. 1 bottom middle), we observe that DoUDP is still significantly more reliable in comparison with DoTCP across all continents for *Probe* resolvers.

Takeaway: Overall, we find that across nearly all continent and Public resolver pairings, DoTCP exhibits a roughly similar failure rate in comparison with DoUDP. However, not all resolver PoPs of a Public resolver support DoTCP universally, resulting in different failure reasons. As for *Probe* resolvers, we observe that failure rates over DoTCP are much higher on each continent, ranging from roughly 63% to 78%, whereas DoUDP is much more reliable for *Probe* resolvers.

By Autonomous System. We further study the failure rates for the largest 10 ASes (based on number of RA probes), i.e., we group the samples by resolver and AS before calculating the failure rates. Fig. 1 shows the failure rates over DoTCP by AS (top right) for the subset of 639 probes hosted in the top 10 ASes, along with the failure rate difference in percentage points to DoUDP (bottom right). Note that due to the deployment of RA probes (see § 3.1), the top 10 ASes are inherently centered around EU and NA.

Overall, we observe that failure rates are around <1–3% for most AS-Public resolver pairings. Similarly, the differences to failure rates over DoUDP are <1% for most, indicating that both DoTCP and DoUDP resolvers work fairly reliably with most *Public* resolvers.

The ASes themselves show comparable failure rates for most *Public* resolvers, while Neustar and UncensoredDNS exhibit increased failure rates ranging from 5.1% up to 28.5% for almost all ASes. In contrast, we notice some pairings that show significantly higher failure rates: For instance, we find outliers in failure rates of roughly 68–69% for DoTCP requests from DTAG, VODANET, and IBSNAZ to Comodo. As 95% of all failed DoTCP measurements to Comodo are *TUCONNECT* errors, RA probes from those ASes are unable to reliably establish TCP connections with Comodo’s recursive resolvers. This is also reflected in the bottom plot, where the increases of 64.9–67.5 percentage points for the same ASes and Comodo show that failures are much less common over DoUDP.

Moreover, probes hosted in the ASes of COMCAST and ORANGE experience even higher failure rates with 97.9% and 100.0%, respectively, towards OpenNIC; probes in the ATT AS also show

moderately high failure rates of 30.9%. This indicates that nearly all DoTCP requests from the former two ASes encounter issues which lead to no valid DoTCP responses from OpenNIC. We find that 96% of all failed DoTCP measurements to OpenNIC result in *Timeout* errors which surpass the 5 second threshold. Considering that other *Public* resolvers and ASes do *NOT* show similarly high failure rates for OpenNIC, this observation suggests issues specific to the paths between OpenNIC and the ASes of COMCAST as well as ORANGE, e.g., blackholing. This is supported by the fact that the differences in failure rates for OpenNIC (Fig. 1 bottom) are 0.0% for both COMCAST and ORANGE, stating that the failure rates are equal using DoTCP and DoUDP.

For *Probe* resolvers, we observe failure rates ranging from 54.4% for COMCAST up to 91.3% for VODANET across all ASes. In contrast to DoUDP, failure rates are much higher, as the differences in percentage points shown in the bottom plot are about equally as high, ranging from 53.3 to 90.2 percentage points. Given that the *Probe* resolver failure rates are even higher than seen in the continent-level analysis, this observation supports our above hypothesis which states that most of the measurements to *Probe* resolvers are forwarded from CPE devices to ISP resolvers with lacking DoTCP support.

In some cases, we find DoTCP to be more reliable than DoUDP: While UncensoredDNS exhibits moderate failure rates (between 8.0% and 28.5%) across all ASes as outlined above, we notice that the failure rate difference to DoUDP ranges from -9.1 to -31.8 percentage points. As such, DoTCP is more reliable than DoUDP for the top 10 ASes when using UncensoredDNS, with most of the DoTCP failures being related to *TUCONNECT* errors (92%). The same pattern applies to Yandex, which shows failure rates from 0.5% to 6.3% for the different ASes over DoTCP, whereas the differences to DoUDP are between -0.9 and -3.6 percentage points, meaning that DoTCP is more reliable than DoUDP for Yandex. Similarly, Quad9 shows failure rate differences of roughly -7 percentage points for the DTAG, VODANET, and IBSNAZ ASes in particular, which means moderately high failure rates of 8–11% over DoUDP but much more reliable DoTCP behavior with 1.8–4.1% failure rates for these ASes. In contrast, Neustar samples show varying failure rates over DoTCP ranging from 0.5% to 19.6%, as well as failure rate differences to DoUDP between -9.1 and $+12.1$ percentage points.

Takeaway: *In our AS-based analysis, we find that failure rates over DoTCP for most pairings of ASes and Public resolvers are low ($<1-3\%$), which roughly matches the respective failure rates over DoUDP (difference around 0 percentage points). However, we also observe cases in which failure rates over DoTCP are much higher: For some ASes, DoUDP requests are mostly successfully responded to by Comodo; yet, the DoTCP requests failed in more than two out of three measurements from these ASes (68.2–69.3% failure rates). Moreover, our observation also indicate paths specific issues between OpenNIC and COMCAST as well as ORANGE, showing failure rates of 97.9% and 100.0% for DoTCP AND DoUDP. Regarding Probe resolvers, we again observe high failure rates over DoTCP (54.4–91.3%) across all top 10 ASes, indicating that Probe resolvers still lack reliable and vast support for DoTCP.*

5 RESPONSE TIMES

To enable a direct comparison of DoTCP and DoUDP, we only include probe:resolver pairs with both successful DoTCP AND DoUDP measurements (see Table 3) in our *Response Times* analysis. Moreover, please recall that domain names are explicitly resolved recursively, which ensures uncached DNS responses and enables comparable results for *Public* and *Probe* resolvers (see § 3).

The *Response Time* is defined as the time between the moment the first packet of the measurement is sent by the RA probe until the moment it receives a valid DNS response. While the first packet for DoUDP is the actual DNS query, the TCP 3-way handshake SYN is the first packet for DoTCP. Since we ensure uncached responses, the *Response Time* also includes the time required for the lookup of the requested domain on the authoritative resolver, and therefore comprises in detail of:

- (1) In case of DoTCP:
 - Connection Establishment: Probe \leftrightarrow Recursive
- (2) Request: Probe \rightarrow Recursive \rightarrow Authoritative
- (3) Response: Authoritative \rightarrow Recursive \rightarrow Probe

Hence, DoTCP requires an additional Round-Trip Time (RTT) for the connection establishment between Probe and recursive resolver (step (1)). Thus, we expect DoTCP to result in higher *Response Times* compared to DoUDP. Therefore, the *Response Times* over DoTCP presented in Fig. 2 (top) resemble the time required for all steps (1)–(3). Moreover, the *Response Time differences* between DoTCP and DoUDP shown in Fig. 2 (bottom) represent the overhead caused by the TCP connection establishment (step (1)), regardless of whether a cached or uncached record is looked up, as the calculated *differences* essentially nullify steps (2)–(3) altogether.

By Continent. We evaluate the observed DNS response times by calculating the median response times per resolver and continent based on the median response times of each probe:resolver pair as shown in Fig. 2 (top middle).

We observe that DoTCP response times vary between resolvers and continents, where Neustar performs considerably worse on each individual continent ranging from 562.5ms in SA to 1,163.7ms in Africa (AF). In contrast, Google does offer the fastest response times over all continents with 71.3ms, as well as on each individual continent. Moreover, we find regional differences over all resolvers, where EU (58.9–309.4ms) and NA (98.4–335.4ms) show considerably faster response times over all resolvers except Neustar. Evaluating AF, we observe that the continent shows the slowest response times for 8 of the 10 *Public* resolvers ranging from 296.5ms (Google) to 1,163.7ms (Neustar), hinting at fewer PoPs in AF. To check this hypothesis, we lookup information on the DNS infrastructures published by the operators of the *Public* resolvers [10–12, 21, 40–43, 54, 59]; for most resolvers, we find that the number of PoPs in AF is indeed lower than in other continents.

To compare DoTCP to DoUDP response times, Fig. 2 (bottom middle) shows the response times *difference* between DoTCP and DoUDP for all resolvers and continents. Positive values (colored in red) indicate higher response times for DoTCP, and negative values (colored in blue) represent higher response times over DoUDP.

In total, the response times increase moderately when using DoTCP instead of DoUDP, ranging from 16.7ms (i.e., an increase by 21.0%, Cloudflare) to 48.8ms (66.6%, Comodo) for all *Public* resolvers

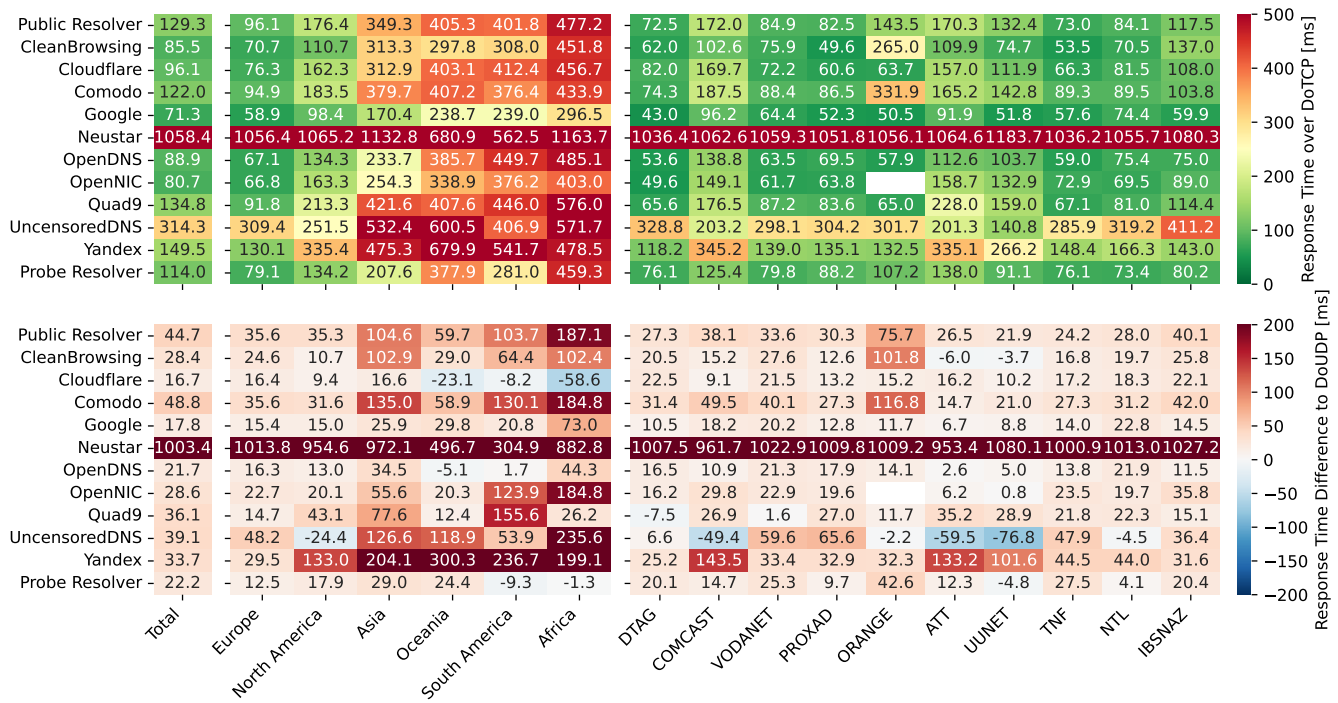


Figure 2: Median response time by Resolver based on medians for probe and resolver over DoTCP (top), along with respective response time difference between DoTCP and DoUDP (bottom); across all samples in total (left), by Continent (middle), and by Top 10 ASes (right), each in descending order by number of probes. Positive values (colored in red) indicate higher median response times for DoTCP.

except Neustar, where the response time is increased by 1,003.4ms (1,824.6%). Overall, the relative increase is less than 37% for 6 out of 10 *Public* resolvers. Cloudflare does show minor increases with 16.6ms for AS, 16.4ms for EU, and 9.4ms for NA, but does manage to achieve lower response times over DoTCP in comparison with DoUDP in AF (−58.6ms), Oceania (OC) (−23.1ms), and SA (−8.2ms) as well. Notably, *Probe* resolvers achieve lower response times for DoTCP in SA (−9.3ms) and AF (−1.3ms), and also show only minor increases of 12.5–29.0ms for the remaining continents. This results in a total increase of 22.2ms for all *Probe* resolvers, which is a relative increase of 24.1% when switching from DoUDP to DoTCP. As *Probe* resolvers primarily consist of ISP resolvers (see § 3.1 and § 4), they are located closer to the *home* probes than *Public* resolvers which are typically hosted in data centers farther away [19]. Therefore, we attribute the lower observed response times to these shorter paths, which result in lower latencies due to faster handshakes.

Takeaway: Response times over DoTCP are highly varying depending on the continent of the probe for *Public* and *Probe* resolvers, ranging from 58.9ms to more than one second. Response times are especially high in AF (296.5–1,163.7ms), which we attribute to the lower number of resolver PoPs in AF. On the other hand, we observe the lowest response times for EU and NA, which are both continents with the most PoPs w.r.t. both resolver endpoints and RA probes. Nevertheless, we find DoTCP to be slower than DoUDP for nearly all pairs of continent and resolver, with largely varying response time differences (from −58.6ms to 1,013.8ms). However, the relative increase over all

continents is less than 25% for all *Probe* resolvers, and less than 37% for 6 out of 10 *Public* resolvers.

By Autonomous System. To investigate response times for the top 10 ASes, we calculate the median response times per resolver and AS based on the median response times of the probe:resolver pairs, i.e., analogous to the continent-based analysis. Fig. 2 presents the absolute medians for each resolver and AS (top right), as well as the difference in response times to DoUDP (bottom right). Note that due to the 100.0% failure rate observed for ORANGE using OpenNIC (see § 4), the respective value could not be determined, which is denoted by the empty cell in the plot.

Overall, the response times roughly match the continent-based response time for EU and NA as discussed in the previous analysis, however, recall that the top 10 ASes are centered around EU and NA. Other patterns, such as Neustar and UncensoredDNS showing higher response times overall, also apply to the AS-based analysis. Similarly, the determined response times are comparable across most resolvers for an individual AS, although we also see outliers with higher response times:

For instance, the NA-based ASes, namely COMCAST, ATT, and UUNET exhibit higher response times of mostly above 100ms, especially for Yandex (266.2–345.2ms). This is likely because PoPs of Yandex’ DNS service are located in Russia, Commonwealth of Independent States (CIS) countries, and Western Europe [59]: Therefore, probes of NA-based ASes are located much farther away than probes in EU, resulting in significantly higher response times.

In contrast, the remaining ASes, all of which are EU-based, measure response times of approximately 50–90ms for most pairings. This is likely due to larger geographical distances that packets have to travel in NA, compared to the more compact landmass in EU, which is ultimately reflected in the ASes as well. An exception to this is IBSNAZ (based in Italy), which shows higher response times of around 100ms for most resolvers, although this is still lower than the response times seen by the NA-based ASes. On the other hand, samples from ORANGE also show higher response times when querying CleanBrowsing (265.0ms) and Comodo (331.9ms).

Contrasting the response times of *Probe* and *Public* resolvers, the AS-based analysis reveals that *Probe* resolvers are mostly on par with *Public* resolvers, whereas the failure rates are significantly worse (see § 4).

Regarding the response time differences between DoTCP and DoUDP, we observe that for most pairings DoTCP is slower by up to 50ms, with many values accumulating around 10–20ms. However, we find that UncensoredDNS achieves response time differences between –49.4ms and –76.8ms for the US-based ASes, indicating that responses from UncensoredDNS over DoTCP are faster than over DoUDP for these ASes. Nevertheless, the exceptional cases of higher response times over DoTCP for specific AS-resolver pairings (discussed above) are all reflected through higher deltas in the difference plot as well, overall showing that DoTCP is slower than DoUDP for probes in the top 10 ASes.

Takeaway: *The top 10 ASes provide a more in-depth perspective of the response time analysis for probes in EU and NA: We notice that NA-based ASes are slow when using Yandex due to geographical distance, as Yandex primarily operates around Russia. On the flip side, EU-based ASes measure the lowest response times, which we attribute to the high density of both resolver PoPs and probe deployments. Altogether, we find Probe resolvers exhibit roughly comparable response times to Public resolvers. Nevertheless, across each AS and nearly all resolvers, DoTCP is slower than DoUDP.*

Optimizations. To put our results into perspective, we take a closer look at two key features aiming to improve DoTCP response time, which are both recommended by recent standardization efforts focussing on operational requirements for DoTCP [33]: *TFO* and *EDNS0 TCP keepalive*. While *TFO* [9] reduces the handshake time by one RTT for connections following an initial exchange of a TFO cookie, *EDNS0 TCP keepalive* [58] allows DNS resolvers to keep a TCP connection alive. Both mechanisms can be leveraged to reduce the DoTCP response time by one RTT, hence, bringing it on par with DoUDP. Although RA does not provide information on the usage of *TFO* or *EDNS0 TCP keepalive* within their documentation (or the measurement results itself), we setup a recursive resolver to explicitly check the probe’s support for both features. For this, we randomly selected 50 probes and issued one DoTCP query per probe. We find that *none* of the requests include either the *TFO* or the *EDNS0 TCP keepalive* option. As RA probes issue DNS measurements identically with the same options and parameters, we conclude that none of our requests used the features.

To evaluate the general support of these features on the *Public* resolvers, we manually check each of the *Public* resolvers by explicitly requesting a TFO cookie and setting the `edns-tcp-keepalive` *EDNS(0)* option in our queries from a single vantage point. However,

none of the tested resolvers returned the `edns-tcp-keepalive` *EDNS(0)* option required for *EDNS0 TCP keepalive*, nor a TFO cookie required for *TFO*. An exception to this is Google, which responds with a TFO cookie. Using the cookie in subsequent connections, however, was only successful in rare cases: The resolver terminated the connection upon receiving the TFO cookie for most measurements, falling back to a traditional TCP handshake instead. This behavior is also observed by [16] and indicates DNS load balancing, which does not factor previous connections into account for server selection. While [33] recommends the usage of TFO by leveraging the same TFO key in load balancing scenarios as well, the behavior as observed on Google actually increases the response time by one RTT due to the connection reset following the refused TFO cookie.

Takeaway: *None of the tested public resolvers support EDNS0 TCP keepalive, and TFO is only supported by Google. However, using the TFO cookie in subsequent connections to Google was only successful in rare cases: due to the connection reset following the refused TFO cookie, the usage of TFO on Google actually increases the response time in the majority of cases.*

6 LIMITATIONS & FUTURE WORK

We acknowledge that observations from RA probes are not fully representative and unconditionally generalizable for the whole Internet. Given probes are mainly deployed in EU and NA, the probes selected in our study are also heavily centered around these regions (see § 3.1). However, we still aggregate *all* samples across a continent (and AS) for the *Failure Rate* and *Response Time* analyses: Reducing the number of samples from EU and NA to be comparable to other continents would have overall resulted in a much lower number of data points and, therefore, in a reduced representativeness of the measurement study.

Since we measure DoTCP from the Edge, we cannot control or analyze possible path influences on the signaled *EDNS(0) Buffer Sizes* (see § 3.2). For instance, middleboxes might change the *EDNS(0) Buffer Size* based on a static configurations or a discovered path MTU. RFC 6891 [14] explicitly prohibits this for simple DNS forwarders. However, the RFC makes an exception for middleboxes with additional functionality, which are allowed to process and act on the *EDNS(0) Buffer Size*; e.g., *CoreDNS* [13] leverages this to override the buffer size in order to prevent *IP fragmentation*.

Additionally, we cannot determine the origin of failures (see § 4) and high delays along the paths (see § 5) in more detail, as RA probes do not provide such information in the measurement results. We also acknowledge that *Probe* resolvers with private IPv4 addresses (e.g., CPEs) may use either ISP or public DNS services as upstream DNS, which we cannot further differentiate with the measured data (see § 4).

As this paper provides a unique insight at DNS over TCP, it is also limited to a View from the Edge. Hence, we plan to extend our study in order to obtain a more complete picture: First, we intend to measure domains for which we operate the authoritative server, which will allow us to have more fine-grained control regarding measured properties like *EDNS(0) Buffer Sizes* and the actual size of DNS responses. While this will contribute a view of DoTCP between resolvers and authoritative servers, we will also explicitly study *truncation* and *IP fragmentation* on stub to recursive resolvers

by issuing DNS queries to controlled recursive resolvers. Moreover, this setup will enable us to study the benefits of *TFO* and *EDNS0 TCP keepalive*, and their effects on application layer protocols. Since QUIC, as a reliable, end-to-end encrypted transport protocol, is designed to improve on several shortcomings of TCP, DNS over QUIC (DoQ) can potentially obsolete the necessity to fall back to DoTCP altogether. We recently presented a study on DoQ [31], where we focused on a comparative analysis between DoQ, DoT, DoH, DoUDP, as well as DoTCP. However, the study is limited to a single vantage point, which is why we plan to measure DoQ from the Edge in order to draw additional comparisons with DoTCP. This will allow us to provide a more holistic view of DNS variations to complement comparable studies on DoT and DoH.

7 CONCLUSION

We presented a unique view on DNS over TCP (DoTCP) from the Edge using 2,500 RIPE Atlas (RA) probes deployed in residential home networks around the globe. Based on 12.1M DNS requests issued to *Public* and *Probe* recursive resolvers over DNS over UDP (DoUDP) and DoTCP, we evaluated *Response Times* as well as *Failure Rates* of DoTCP.

We showed that *Response Times* are highly varying depending on the continent of the probes location for *Public* and *Probe* resolvers, and that DoTCP is generally slower than DoUDP. Although this was expected, the relative increase in *Response Time* is less than 37% for most resolvers. While *TFO* and *EDNS0 TCP keepalive* can be leveraged to further reduce the DoTCP response times, we showed that support on *Public* resolvers is still missing, hence leaving room for optimizations in the future.

Analyzing *Failure Rates*, we determined that *Public* resolvers generally have comparable reliability for DoTCP and DoUDP. However, *Probe* resolvers show a significantly different behavior, as their failure rate for DoTCP is considerably higher with 74.96%; as a result, DoTCP queries targeting *Probe* resolvers fail in 3 out of 4 cases. Therefore, *Probe* resolvers largely do not comply with the standard described in RFC 7766, which states that all DNS implementations MUST support both DoUDP as well as DoTCP. As such, *Probe* resolvers face issues with fallback to DoTCP in case of large DNS responses to date. This problem will only aggravate in the future: As DNS response sizes will continue to grow, the need for DoTCP will solidify.

ACKNOWLEDGEMENTS

We thank Jan R uth, Alessandro Finamore, Matteo Varvello, and the anonymous reviewers for their insightful feedback and guidance. In addition, we thank RIPE Atlas for providing the measurement infrastructure. This work was partly supported by the Volkswagenstiftung Nieders achsisches Vorab (Funding No. ZN3695).

REFERENCES

- [1] Alexa. 2022. Top sites. <https://www.alexa.com/topsites> [Online; accessed 2022-Mar-31].
- [2] Vaibhav Bajpai et al. 2015. Lessons Learned From Using the RIPE Atlas Platform for Measurement Research. *Computer Communication Review* 45, 3 (2015), 35–42. <https://doi.org/10.1145/2805789.2805796>
- [3] Vaibhav Bajpai et al. 2019. The Dagstuhl Beginners Guide to Reproducibility for Experimental Networking Research. *SIGCOMM Comput. Commun. Rev.* 49, 1 (Feb. 2019), 24–30. <https://doi.org/10.1145/3314212.3314217>

- [4] BIND9. 2022. BIND 9.16.8 Release Notes. https://bind9.readthedocs.io/en/v9_16_8/notes.html#notes-for-bind-9-16-8 [Online; accessed 2022-Mar-31].
- [5] Ron Bonica et al. 2020. IP Fragmentation Considered Fragile. RFC 8900. <https://doi.org/10.17487/RFC8900>
- [6] Robert T. Braden. 1989. Requirements for Internet Hosts - Application and Support. RFC 1123. <https://doi.org/10.17487/RFC1123>
- [7] Markus Brandt et al. 2018. Domain Validation++ For MitM-Resilient PKI. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. <https://doi.org/10.1145/3243734.3243790>
- [8] Sebastian Castro et al. 2010. Understanding and Preparing for DNS Evolution. In *Traffic Monitoring and Analysis*. https://doi.org/10.1007/978-3-642-12365-8_1
- [9] Yuchung Cheng et al. 2014. TCP Fast Open. RFC 7413. <https://doi.org/10.17487/RFC7413>
- [10] CleanBrowsing. 2022. How CleanBrowsing Works. <https://cleanbrowsing.org/how-it-works/> [Online; accessed 2022-Mar-31].
- [11] Cloudflare. 2022. The Cloudflare global networks. <https://www.cloudflare.com/network/> [Online; accessed 2022-Mar-31].
- [12] Comodo. 2022. Appendix 2 - Comodo Secure DNS Service. <https://help.comodo.com/topic-399-1-904-11738-Appendix-2---Comodo-Secure-DNS-Service.html> [Online; accessed 2022-Mar-31].
- [13] CoreDNS. 2022. CoreDNS bufsize plugin. <https://github.com/coredns/coredns/tree/master/plugin/bufsize> [Online; accessed 2022-Mar-31].
- [14] Joao da Silva Damas et al. 2013. Extension Mechanisms for DNS (EDNS(0)). RFC 6891. <https://doi.org/10.17487/RFC6891>
- [15] Tianxiang Dai et al. 2021. DNS-over-TCP Considered Vulnerable. In *Proceedings of the Applied Networking Research Workshop (ANRW '21)*, 76–81. <https://doi.org/10.1145/3472305.3472884>
- [16] Casey Deccio and Jacob Davis. 2019. DNS Privacy in Practice and Preparation (CoNEXT '19). <https://doi.org/10.1145/3359989.3365435>
- [17] John Dickinson et al. 2016. DNS Transport over TCP - Implementation Requirements. RFC 7766. <https://doi.org/10.17487/RFC7766>
- [18] DNS flag day. 2020. DNS flag day 2020. <https://dnsflagday.net/2020/> [Online; accessed 2022-Mar-31].
- [19] Trinh Viet Doan et al. 2021. Evaluating Public DNS Services in the Wake of Increasing Centralization of DNS. In *IIFIP Networking 2021*. <https://doi.org/10.23919/IIFIPNetworking52078.2021.9472831>
- [20] Kazunori Fujiwara and Paul A. Vixie. 2021. *Fragmentation Avoidance in DNS*. Internet-Draft draft-ietf-dnsop-avoid-fragmentation-05. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-avoid-fragmentation-05> Work in Progress.
- [21] Google. 2022. Google Public DNS: Frequently Asked Questions. <https://developers.google.com/speed/public-dns/faq#locations> [Online; accessed 2022-Mar-31].
- [22] Amir Herzberg and Haya Shulman. 2013. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *2013 IEEE Conference on Communications and Network Security*. <https://doi.org/10.1109/CNS.2013.6682711>
- [23] Paul E. Hoffman, Andrew Sullivan, and Kazunori Fujiwara. 2019. DNS Terminology. RFC 8499. <https://doi.org/10.17487/RFC8499>
- [24] Thomas Holterbach et al. 2015. Quantifying Interference between Measurements on the RIPE Atlas Platform. In *Internet Measurement Conference 2015*. <https://doi.org/10.1145/2815675.2815710>
- [25] Austin Hounsel et al. 2020. Comparing the Effects of DNS, DoT, and DoH on Web Performance. In *Proceedings of The Web Conference 2020*. <https://doi.org/10.1145/3366423.3380139>
- [26] Geoff Huston. 2020. Measuring the impact of DNS Flag Day 2020. <https://blog.apnic.net/2020/12/14/measuring-the-impact-of-dns-flag-day-2020/> [Online; accessed 2022-Mar-31].
- [27] ICANN. 2022. RSSAC002 Data. <https://www.dns.icann.org/rssac/rssac002/> [Online; accessed 2022-Mar-31].
- [28] IETF. 1987. Domain names - concepts and facilities. RFC 1034. <https://doi.org/10.17487/RFC1034>
- [29] IETF. 1987. Domain names - implementation and specification. RFC 1035. <https://doi.org/10.17487/RFC1035>
- [30] Axel Koolhaas and Tjeerd Slokkker. 2020. *Defragmenting DNS*. <https://homepages.staff.os3.nl/~delaat/rp/2019-2020/p78/report.pdf>
- [31] Mike Kosek et al. 2022. One to Rule Them All? A First Look at DNS over QUIC. In *Passive and Active Measurement 2022*. https://doi.org/10.1007/978-3-030-98785-5_24
- [32] Christian Kreibich et al. 2010. Netalyzr: Illuminating the Edge Network. In *Internet Measurement Conference 2010*. <https://doi.org/10.1145/1879141.1879173>
- [33] John Kristoff and Duane Wessels. 2022. DNS Transport over TCP - Operational Requirements. RFC 9210. <https://doi.org/10.17487/RFC9210>
- [34] Vladimir Ksinant et al. 2003. DNS Extensions to Support IP Version 6. RFC 3596. <https://doi.org/10.17487/RFC3596>
- [35] NLnet Labs. 2006. Measuring the effects of DNSSEC deployment on query load. <http://www.nlnetlabs.nl/downloads/publications/dnssec/dnssec-effects.pdf> [Online; accessed 2022-Mar-31].

- [36] Chaoyi Lu et al. 2019. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In *Internet Measurement Conference 2019*. <https://doi.org/10.1145/3355369.3355580>
- [37] Jiarun Mao et al. 2022. Assessing Support for DNS-over-TCP in the Wild. In *Passive and Active Measurement 2022*. https://doi.org/10.1007/978-3-030-98785-5_22
- [38] Giovane C. M. Moura et al. 2020. Clouding up the Internet: How Centralized is DNS Traffic Becoming?. In *Internet Measurement Conference 2020*. <https://doi.org/10.1145/3419394.3423625>
- [39] Giovane C. M. Moura et al. 2021. Fragmentation, Truncation, and Timeouts: Are Large DNS Messages Falling to Bits?. In *Passive and Active Measurement 2021*. https://doi.org/10.1007/978-3-030-72582-2_27
- [40] Neustar. 2022. Cloud-based Authoritative DNS for Faster, Safer and Accurate Query Resolution. https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/product-literature/dns/ultradns-overview.pdf [Online; accessed 2022-Mar-31].
- [41] OpenDNS. 2022. Data Center Locations. <https://www.opendns.com/data-center-locations/> [Online; accessed 2022-Mar-31].
- [42] OpenNIC. 2022. Tier-2 Server History. <http://report.opennic.org/> [Online; accessed 2022-Mar-31].
- [43] Quad9. 2022. Locations. <https://www.quad9.net/service/locations/> [Online; accessed 2022-Mar-31].
- [44] Audrey Randall et al. 2020. Trufflehunter: Cache Snooping Rare Domains at Large Public DNS Resolvers. In *Internet Measurement Conference 2020*. <https://doi.org/10.1145/3419394.3423640>
- [45] Eric Rescorla et al. 2021. *TLS Encrypted Client Hello*. Internet-Draft draft-ietf-tls-esni-13. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-13> Work in Progress.
- [46] RIPE NCC. 2015. RIPE Atlas: A Global Internet Measurement Network. In *Internet Protocol Journal 2015*. <http://ipj.dreamhosters.com/wp-content/uploads/2015/10/ipj18.3.pdf>
- [47] RIPE NCC. 2022. RIPE Atlas Docs. https://beta-docs.atlas.ripe.net/apis/measurements/types/type_specific_attributes.html#dns [Online; accessed 2022-Mar-31].
- [48] RIPE NCC. 2022. RIPE Atlas Service Terms and Conditions Version 3.4. <https://atlas.ripe.net/legal/terms-conditions/> [Online; accessed 2022-Mar-31].
- [49] Scott Rose et al. 2005. Resource Records for the DNS Security Extensions. RFC 4034. <https://doi.org/10.17487/RFC4034>
- [50] Kyle Schomp et al. 2013. On Measuring the Client-Side DNS Infrastructure. In *Internet Measurement Conference 2013*. <https://doi.org/10.1145/2504730.2504734>
- [51] Benjamin M. Schwartz et al. 2021. *Service binding and parameter specification via the DNS (DNS SVCB and HTTPS RRs)*. Internet-Draft draft-ietf-dnsop-svcb-https-07. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-svcb-https-07> Work in Progress.
- [52] Trinh Viet Doan et al. 2021. Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times. In *Passive and Active Measurement 2021*. https://doi.org/10.1007/978-3-030-72582-2_12
- [53] unbound. 2022. unbound 1.12.0 Release Notes. <https://github.com/NLnetLabs/unbound/releases/tag/release-1.12.0> [Online; accessed 2022-Mar-31].
- [54] UncensoredDNS. 2022. DNS servers. <https://blog.uncensoreddns.org/dns-servers/> [Online; accessed 2022-Mar-31].
- [55] Gijs Van Den Broek et al. 2014. DNSSEC meets real world: dealing with unreachability caused by fragmentation. *IEEE Communications Magazine* 52, 4 (2014), 154–160. <https://doi.org/10.1109/MCOM.2014.6828880>
- [56] Paul A. Vixie. 1999. Extension Mechanisms for DNS (EDNS0). RFC 2671. <https://doi.org/10.17487/RFC2671>
- [57] Nicholas Weaver et al. 2011. Implications of netalyzrs dns measurements. In *in Proceedings of the First Workshop on Securing and Trusting Internet Names 2011*. <http://www.icsi.berkeley.edu/pubs/networking/implicationsofnetalyzrsdns11.pdf>
- [58] Paul Wouters et al. 2016. The edns-tcp-keepalive EDNS0 Option. RFC 7828 (2016), 1–11. <https://doi.org/10.17487/RFC7828>
- [59] Yandex.DNS. 2022. Technical details. <https://dns.yandex.com/advanced/> [Online; accessed 2022-Mar-31].

- The file `analysis.ipynb` is a jupyter notebook containing all analyses detailed in the paper.
- The supplementary file `public-resolvers-ipv4s.csv` is single column text file containing a list of known public resolvers (used in related work).
- The supplementary file `pyasn.dat` is a 2 columns text file mapping RIPEAtlas (RA) probes IP address to the related ASN.
- The file `measurements.parquet` contains the full measurements campaign run via RA probes.

Analysis

- Open the Jupyter Notebook `analysis.ipynb`.
- Run the Jupyter Notebook. Depending on machine capabilities, this can take from several minutes up to a few hours for the full dataset.

APPENDIX - REPRODUCIBILITY

In order to enable the reproduction of our findings, we make the raw data of our measurements as well as the analysis scripts and supplementary files publicly available on GitHub:

<https://github.com/kosekmi/2022-ccr-dns-over-tcp-from-the-edge>

This section gives an overview over the contents of the repository. More details are provided in the `README.md` within the GitHub repository.

Repository Overview