One to Rule Them All? A First Look at DNS over QUIC

Mike Kosek[1](✉), Trinh Viet Doan[1], Malte Granderath[1], and Vaibhav Bajpai[1,2]

[1] Technical University of Munich, Munich, Germany
{kosek,doan,grandera}@in.tum.de
[2] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
bajpai@cispa.de

Abstract. The DNS is one of the most crucial parts of the Internet. Since the original DNS specifications defined UDP and TCP as the underlying transport protocols, DNS queries are inherently unencrypted, making them vulnerable to eavesdropping and on-path manipulations. Consequently, concerns about DNS privacy have gained attention in recent years, which resulted in the introduction of the encrypted protocols DNS over TLS (DoT) and DNS over HTTPS (DoH). Although these protocols address the key issues of adding privacy to the DNS, they are inherently restrained by their underlying transport protocols, which are at strife with, e.g., IP fragmentation or multi-RTT handshakes—challenges which are addressed by QUIC. As such, the recent addition of DNS over QUIC (DoQ) promises to improve upon the established DNS protocols. However, no studies focusing on DoQ, its adoption, or its response times exist to this date—a gap we close with our study. Our active measurements show a slowly but steadily increasing adoption of DoQ and reveal a high week-over-week fluctuation, which reflects the ongoing development process: As DoQ is still in standardization, implementations and services undergo rapid changes. Analyzing the response times of DoQ, we find that roughly 40% of measurements show considerably higher handshake times than expected, which traces back to the enforcement of the traffic amplification limit despite successful validation of the client's address. However, DoQ already outperforms DoT as well as DoH, which makes it the best choice for encrypted DNS to date.

1 Introduction

The Domain Name System (DNS) is used for almost all communications across the Internet. As the original DNS specifications [25,26] define UDP and TCP as the underlying transport protocols, DNS requests and responses using DNS over UDP (DoUDP) and DNS over TCP (DoTCP) are inherently unencrypted, which makes them vulnerable to eavesdropping and on-path manipulations [56]. This enables an observer to not only reveal the browsing or application usage behavior [57], but also the identification of device types which are in use [35];

O. Hohlfeld et al. (Eds.): PAM 2022, LNCS 13210, pp. 537–551, 2022.
https://doi.org/10.1007/978-3-030-98785-5_24

hence, a user profile can be created and tracked with only having access to the user's DNS traffic [44,45,48]. Consequently, concerns on DNS privacy have gained attention in recent years.

With the standardization of DNS over TLS (DoT) [38] in 2016 and DNS over HTTPS (DoH) [37] in 2018, encrypted DNS protocols leveraging Transport Layer Security (TLS) on top of TCP have been introduced. Moreover, DNS over DTLS (DoDTLS) [52] has also been standardized as an experimental protocol in 2017, offering encrypted DNS by leveraging TLS on top of UDP.

However, while these protocols address the key issues of adding privacy to DNS [29,33,49,55], they are inherently restrained by their underlying transport protocols. Using UDP as a connectionless protocol, DoDTLS is vulnerable to *IP fragmentation* [28,31,36,50]—a problem which has gained awareness in recent years due to the trend of increasing DNS response sizes [40,46]. Although both DoT and DoH are not affected by *IP fragmentation*, as they leverage the connection-oriented TCP, the underlying TCP connections are still constrained by head-of-line-blocking and missing multiplexing support on the transport layer, as well as an additional connection establishment in comparison to UDP.

These challenges are addressed by QUIC [42,43,54], a connection-oriented encrypted transport protocol using UDP as a substrate. Standardized in early 2021, QUIC features mandatory encryption, solves head-of-line blocking, provides multiplexing, and improves on connection establishment time by combining the transport and encryption handshakes into a single round trip. Consequently, offering DNS using the QUIC transport protocol is the natural evolution for not only the traditional performance-oriented DNS protocols DoUDP and DoTCP, but also the privacy-preserving DNS protocols DoT and DoH (as well as the experimental DoDTLS).

DNS over QUIC (DoQ) is currently being standardized within the *DNS PRIVate Exchange* IETF working group [41] with the design goal to provide DNS privacy with minimum latency. With this objective, DoQ aims to obsolete all other currently used DNS protocols, which lack privacy and/or require more round-trips for handshakes—therefore, promising to make DoQ the *"One to Rule them All"*. Despite its development status, multiple experimental implementations already exist that offer DoQ support for clients [7,10,16,19,21], servers [5,7,8,19], proxies [4,16,21], as well as multipurpose libraries [1,7,12,19]. Moreover, *AdGuard* [3] and *nextDNS* [17] already use DoQ in production systems for their DNS-based ad as well as tracker blocking services, offering publicly reachable DoQ servers and client implementations [10,16]. However, while DoQ was submitted to the Internet Engineering Steering Group (IESG) for publication in December 2021, only one study [33] has explicitly included DoQ as part of an experiment on encrypted DNS based on traffic flow analyses as of January 2022. Hence, no studies focusing on DoQ, its adoption, or its response times exist to this date—a gap we close with our study.

We begin by investigating the adoption of DoQ (see Sect. 3) and identify a maximum of 1,217 resolvers in a single week. Over the course of 29 weeks, we find 1,851 unique X.509 certificates used by the resolvers. However, only 51.6%

of the resolvers in the first week are still reachable in the last week, reflecting the ongoing development and standardization process, during which DoQ implementations and services undergo rapid changes. Analyzing the response times of DoQ in comparison to DoUDP, DoTCP, DoT, as well as DoH (see Sect. 4), we find that QUIC's full potential is only utilized in around 20% of measurements. On the other hand, roughly 40% of measurements show considerably higher handshake times than expected, which traces back to the enforcement of the traffic amplification limit despite successful validation of the client's address, ultimately causing an additional, unnecessary round-trip.

The remainder of this paper is structured as follows: We first present our methodology in Sect. 2. Afterwards, we detail our adoption measurements in Sect. 3 before analyzing the response times of DoQ in Sect. 4. Limitations and future work are discussed in Sect. 5, after which we conclude the paper with Sect. 6.

2 Methodology

To study the adoption and response times of DNS over QUIC (DoQ), we issue measurements from a single vantage point located in the research network of the Technical University of Munich, Germany. Distributed measurement platforms such as RIPE Atlas do currently not support DoQ; nevertheless, we plan to distribute our measurements to multiple vantage points in the future (see Sect. 5).

Adoption. To assess the adoption of DoQ on resolvers worldwide, we issue weekly scans of the IPv4 address space over the course of 29 weeks, starting in 2021-W27 (July 05–11). For this, we leverage the *ZMap* network scanner [22] and target all DoQ ports proposed by the different DoQ Internet-Drafts (I-Ds), i.e., `UDP/784`, `UDP/853`, and `UDP/8853` [39]. For comparison, we additionally target DoUDP port `UDP/53`, which we identify by leveraging the *ZMap*'s built-in DNS probing packet that queries an `A` record for `www.google.com` [58]. Since *ZMap* does not provide means for the identification of QUIC or DoQ, we issue a custom packet [24] that carries the `Initial` QUIC handshake frame with an invalid version number of `0` [53]: In this way, if the target operates a QUIC stack on the probed port, a `Version Negotiation` packet is triggered. As such a packet does not produce state, it allows us to identify the target as *QUIC-capable* without consuming resources on the target itself [43]. However, note that other QUIC services, which are not necessarily DoQ, could be offered on the probed ports. Hence, we further validate targets identified as *QUIC-capable* by the *ZMap* scans, checking if they actually support DNS over QUIC [23]. To do so, we offer the `doq` `Application-Layer Protocol Negotiation (ALPN)` identifiers (as required by the DoQ I-Ds [39]), which results in a list of *DoQ-capable* targets. As a final step, a connection to every *DoQ-capable* target on all proposed DoQ ports `UDP/784`, `UDP/853`, as well as `UDP/8853`, is established [15]: For these connections, we offered the QUIC version `draft-34` in our `Initial` frame until 2021-W42, while support for version `1` was added in 2021-W43. Overall, our client supports the QUIC versions `draft-34`, `-32` and `-29` since the start of our study, as well as version 1 later on; hence, the client can respond to Version Negotiation packets if issued by the resolvers. For DoQ, we offer versions in the order of draft-06 to draft-00 [39], for which we added support for new versions within 2 weeks of the draft release. By issuing the highest QUIC and DoQ protocol versions supported by our client first, we ensure that we negotiate the highest shared protocol versions between our client and the target resolver. With this, we record the negotiated QUIC and DoQ versions, as well as the X.509 certificate offered by each *DoQ-capable* target, creating the final list of *DoQ-verified* resolvers.

Response Time. To study the response times of DoQ compared to DoUDP, DoTCP, DoT, and DoH, we develop *DNSPerf*, an open-source DNS measurement tool which supports all stated protocols [11]. Using *DNSPerf* to target all *DoQ-verified* resolvers identified in 2022-W02, we issue response time measurements every hour over the course of 2022-W03 (January 17–23). As we specifically scan for DoQ in our adoption measurement, we measure DoUDP, DoTCP, DoT, as well as DoH *optimistically*, i.e., without prior knowledge whether the target resolvers offer the respective DNS protocols in addition to DoQ. In detail, we measure DoUDP and DoTCP according to RFC 1034 [25] on target port UDP/53 and, respectively, TCP/53, DoT according to RFC 7858 [38] preferring TLS version 1.3 on target port TCP/853, and DoH according to RFC 8484 [37], also preferring TLS version 1.3 on target port TCP/443. Similar to the verification step of the adoption measurement, we again target all proposed DoQ ports.

Our DNS requests query an A record for test.com. We further explicitly set the Recursion Desired flag in all requests to ensure that the resolvers return a valid and recursively queried or cached A record, which circumvents resolvers simply returning the corresponding name server or refusing to answer our queries to prevent *Cache Snooping* [20,51] when the flag is not set. As populating the caches can affect the measured response times, every DNS request on every protocol is preceded by an identical cache warming query, which ensures that the actual DNS response time measurement query is directly answered by the resolver from a cached record. For DoQ, we additionally use the negotiated QUIC Version along with the token received in a New_Token frame of the cache warming query for the handshake of the subsequent DNS response time measurement query, which ensures that the response time measurements are not affected by QUIC's Version Negotiation and Address Validation features. Overall, these decisions enable comparable DNS response time measurements of all stated protocols.

Round-Trip Time (RTT). If the response time measurement of a protocol:resolver pair is successful, we measure the *round-trip time* to the targets to analyze the path and protocol-specific *RTT* [9]. Since the resolvers can be deployed behind proxies, or the path can have protocol-dependent queuing characteristics, we send probing packets to the same port using the same protocol as the respective response time measurement. Therefore, our implementation enables *RTT* measurements for UDP (DoUDP), TCP (DoTCP, DoT, DoH), as well as QUIC (DoQ) by leveraging protocol-specific probing payloads: For UDP, a randomized payload is sent from a random `Source Port`, while the payload for TCP is a `SYN` packet containing a randomized `Sequence Number`. Finally, QUIC leverages the custom packet that carries the `Initial` QUIC handshake with an invalid version number of `0` (see Adoption above).

Ethical Considerations. To minimize the impact of our active scans, we follow best practices of the Internet measurement community [30,47]. Thus, we display the intent of our scans on a website reachable via the IP address of each scanning machine, also allowing targets to opt-out from our study. Moreover, we honor opt-out requests from previous studies and maintain a University-wide shared blocklist with the excluded targets.

Reproducibility. In order to enable the reproduction of our findings [27], we make the developed tools, the raw data of our measurements, as well as the analysis scripts and supplementary files publicly available [18].

3 Adoption

To study the adoption of DoQ on resolvers worldwide, we issue weekly scans of the IPv4 address space over the course of 29 weeks, as described in Sect. 2. Thus, we record the negotiated QUIC and DoQ versions, as well as the X.509 certificates offered by the target resolvers that support DoQ, for which we also determine the announcing Autonomous Systems (ASes) and geolocations. Overall, we find 1,851 unique X.509 certificates over the course of 29 weeks.

Adoption of QUIC and DoQ Versions. In our scans and in the verification process, we target all proposed DoQ ports `UDP/784`, `UDP/853`, and `UDP/8853`. The DoQ drafts `-00` and `-01` state that port `UDP/784` *MAY* be used for experimentation. `draft-02` defined `UDP/8853` for usage as experimentation as well as for reservation at the Internet Assigned Numbers Authority (IANA). This was changed in `draft-03`, where port `UDP/784` was again stated for experimentation usage; ultimately, `UDP/853` has been established as the final port for reservation at IANA. Over the course of the 29 weeks, we observe a dominance of the usage of port `UDP/784`, with roughly 75–82% of all *DoQ-verified* resolvers offering all observed DoQ `drafts-00`, `-02`, and `-03` on `UDP/784`. Port `UDP/8853` is only observed in combination with `draft-02` at roughly 17–24% of all *DoQ-verified* resolvers, with the remainder (<1%) serving DoQ `draft-02` on port `UDP/853`.

Figure 1 presents the *DoQ-verified* resolvers per week, grouped by negotiated DoQ and QUIC version. Overall, we observe that the number of *DoQ-verified* resolvers rises steadily: Starting with 833 resolvers in 2021-W27 (July 05–11), we see an increase by 46.1% to 1,217 verified resolvers in 2022-W03 (January 17–23). After we added support for QUIC version 1 [43] in 2021-W43, we observe a steady usage of `DoQ Draft 02/QUIC 1` (dark blue bars) until 2021-W50, followed by a steep increase until 2022-W01. Analyzing this observation, we find that the open source DNS server implementation *AdGuard Home (AGH)* [5] changed the default DoQ/QUIC pair from `DoQ Draft 02/QUIC Draft 34` (orange bars) to

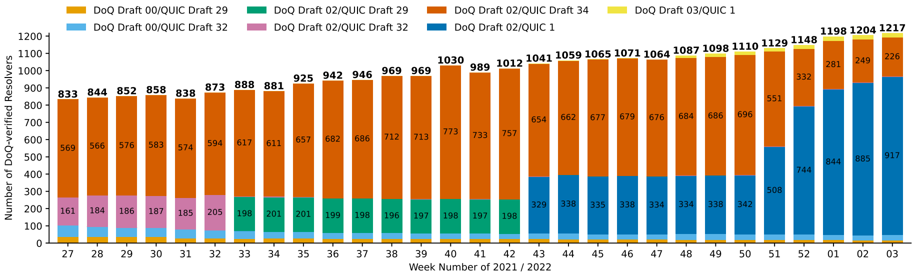


Fig. 1. Number of *DoQ-verified* resolvers per week number of 2021 and 2022 grouped by negotiated DoQ and QUIC version. Support for QUIC version `1` was added in 2021-W43. (Color figure online)

`DoQ Draft 02/QUIC 1` (dark blue bars) starting 2021-W51 [6], matching the pattern we observe. In addition, we find indications of the usage of *AGH* by the updated resolvers within the *Common Names* of their X.509 certificates, and also identify multiple of the updated resolvers to be running *AGH* through random sampling. Hence, we attribute the observed increase in usage of `DoQ Draft 02/QUIC 1` between 2021-W51 and 2022-W01 to this implementation.

Although we offer a total combination of 28 DoQ/QUIC version pairs as of 2022-W03 (see Sect. 2), we observe only 7 pairs across all measurements, with the majority being `DoQ Draft 02/QUIC 1` (dark blue bars, 917 (75.3%)) in 2022-W03. Additionally, we find that only 430 (51.6%) of the initial 833 resolvers are still verified in 2022-W03. As a comparison, 96.5% of the verified DoUDP resolvers from 2021-W27 are still verified in 2022-W03. This fluctuation of DoQ reflects the development process: While DoQ is still in standardization, implementations and services change frequently and are expected to be used in experimental rather than in production environments.

However, both *AdGuard* [3] and *nextDNS* [17] actually do use DoQ in production systems for their DNS-based ad and tracker blocking services, offering publicly reachable DoQ servers as well as client implementations [10,16]. This is reflected in the *Common Names* of the X.509 certificates offered by the verified DoQ resolvers: In 2022-W03, 199 resolvers (16.5%) state `dns.nextdns.io` as their common name. Analyzing the change over time, we observe that *nextDNS* operates the highest share of resolvers in each week, with a mean of roughly 180 resolvers in 2021-W27 to 2021-W31, increasing to a mean of 199 resolvers in 2021-W32 to 2022-W03. While the increase was observed between 2021-W31 and 2021-W32, *nextDNS* offered `DoQ Draft 02/QUIC Draft 32` (purple bars) until 2021-W32 and downgraded all resolvers to `DoQ Draft 02/QUIC Draft 29` (green bars) in 2021-W33, where this DoQ/QUIC pair is exclusively offered by *nextDNS*. After adding support for QUIC version 1 in 2021-W43, we also observe that all *nextDNS* resolvers offer `DoQ Draft 02/QUIC 1` (dark blue bars) since that week; hence, we attribute the previously observed downgrade to the missing support of QUIC version 1 in our tooling during that timeframe. Considering the publicly reachable DoQ servers of *AdGuard* (identified by the common names

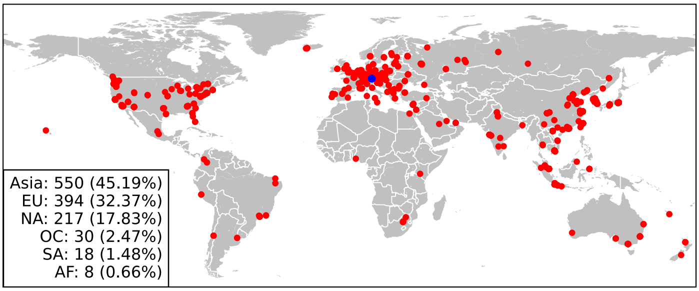


Fig. 2. Geographical locations of the 1,217 *DoQ-verified* resolvers as of 2022-W03, with counts by continent. The blue marker represents our vantage point. (Color figure online)

`dns.adguard.com` and `adguard.ch`), we identify 25 resolvers offering `DoQ Draft 03/QUIC 1` (yellow bars) in 2022-W03 (2.1%). Note that this DoQ/QUIC pair is exclusively offered by the *AdGuard* services, as it differs from the *AdGuard Home (AGH)* open source DNS server implementation detailed above. We find 12–17 resolvers with the common name `dns.adguard.com` and `DoQ Draft 02/QUIC Draft 34` (orange bars) until 2021-W47, after which these resolvers switch to `DoQ Draft 03/QUIC 1` (yellow bars) starting 2021-W48. Moreover, `DoQ Draft 03/QUIC 1` is also offered by 6–8 resolvers using `adguard.ch` starting 2021-W49.

Adoption in Continents and by ASes. Figure 2 presents the geographical locations of the 1,217 *DoQ-verified* resolvers of 2022-W03 with counts per continent based on an IPv4 geolocation lookup service [14]. We observe a strong focus of resolvers operated in Asia (45.19%) and Europe (EU) (32.37%), whereas only 17.83% are operated in North America (NA). However, note that geolocation lookups of IP addresses are known to have inaccuracies, possibly resulting in the incorrect attribution of locations.

The publicly available information of *AdGuard* [2] states that they operate resolvers in 10 countries in the four continents Asia, EU, NA, as well as Oceania (OC). However, this is not reflected in our measurements: We find that 16 resolvers are operated in Russia (EU, MNGTNET (AS199274)), 8 in Cyprus (Asia, ADGUARD (AS212772)), and 1 in Italy (EU, TISCALI-IT (AS8612)) for 2022-W03, resulting in an overall distribution over 2 continents, 3 countries, as well as 3 ASes. Due to the strong divergence, we attribute this observation to the incorrect attribution of the IP geolocation lookups.

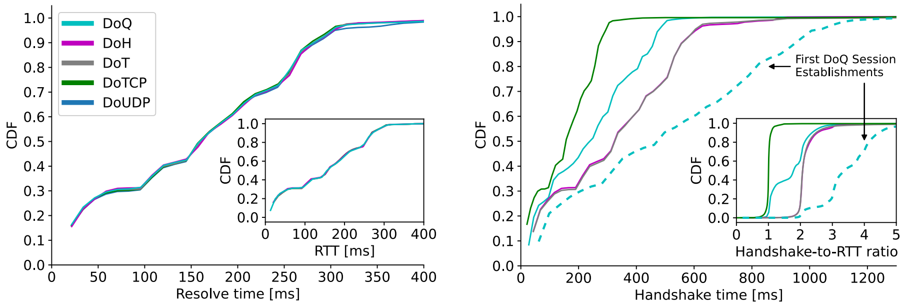
On the other hand, *nextDNS* operates globally distributed DoQ resolvers: The 199 *DoQ-verified* resolvers of 2022-W03 are distributed across 66 countries on all 6 continents, with most resolvers located in EU (78, 39.20%), NA (54, 27.14%), and Asia (35, 17.59%). Looking at the distribution over ASes, we find that 55 (27.64%) are attributed to ANEXIA (AS42473), whose *DoQ resolvers* are all operated by *nextDNS*. The remaining 144 resolvers (72.36%) are distributed over 72 ASes, with most ASes hosting 1–2 resolvers.

While our measurements show a slowly but steadily increasing adoption of DoQ on resolvers worldwide, the observed week-over-week fluctuations reflect the ongoing development and standardization process with rapidly changing implementations and services. Considering that these experimental deployments of the resolvers, along with their geographical locations, can substantially affect the overall response times, in particular when multiple RTTs are required, we investigate the measured response times in the following section.

4 Response Times

To study the response times of DoQ in comparison to DoUDP, DoTCP, DoT, and DoH, we issue response time measurements every hour over the course of 2022-W03 (January 17–23), targeting all *DoQ-verified* resolvers identified in 2022-W02 (see Sect. 2). From these 1,204 resolvers, 1,148 answer our requests via DoQ, 663 via DoH, 1,028 via DoT, 630 via DoTCP, and 455 via DoUDP in 2022-W03. A total of 264 resolvers, i.e., *DoX-verified*, offer all stated DNS protocols simultaneously. While the adoption measurements (Sect. 3) are independent of the selected vantage point, we acknowledge that the vantage point introduces a location bias for our response time measurements (see Sect. 5). To counteract these limitations, we restrict our response time analysis to the 264 *DoX-verified* resolvers, enabling a comparative study of all stated DNS protocols. The geographical distribution of these resolvers follows the distribution observed in the adoption scan: Asia dominates with 123 (46.59%) of the *DoX-verified* resolvers, followed by EU with 83 (31.44%), and NA with 51 (19.32%). The remainder are attributed to OC (5 resolvers, 1.89%) and AF (2 resolvers, 0.76%).

To account for the different transport protocol mechanisms leveraged by the measured protocols, we differentiate between the *round-trip time (RTT)*, the *resolve time*, and the *handshake time*. We define the *resolve time* as the time between the moment the first packet of the DNS query is sent until the moment a valid DNS response is received. Considering we ensure that our requested DNS record is cached by the targeted resolver through cache warming (see Sect. 2), the *resolve time* is, therefore, expected to resemble roughly 1 *RTT* for every measured protocol. In addition to the *resolve time*, we define the *handshake time* as the time between the moment the first packet of the session establishment is sent until the moment the (encrypted) session to the resolver is established. Note that since DoUDP uses a connectionless protocol and, therefore, has no connection establishment, it is omitted from the *handshake time* discussion below. The DoTCP *handshake time* resembles the TCP 3-way handshake, i.e., 1 *RTT*. For DoT and DoH, the TLS handshake is added to the TCP handshake: Using TLS 1.2, the *handshake time* is 3 *RTTs* for DoT and DoH, which is decreased down to 2 *RTTs* with the usage of TLS 1.3. Since QUIC combines the handshakes of the connection as well as the encryption in the `Initial` frame, and we ensure that QUIC's `Version Negotiation` and `Address Validation` features do not affect the actual DNS response time measurement query (see Sect. 2), the *handshake time* of DoQ should resemble 1 *RTT*.



(a) *Resolve time* and *RTT* distribution. (b) *Handshake time* and *handshake-to-RTT ratio* distribution.

Fig. 3. Distributions of response time metrics, targeting 264 *DoX-verified* resolvers. Please note the different x-axis scales. (Color figure online)

Note that we send every request with a new session for every protocol as a single query; we acknowledge that using a previously established session would reduce the overhead introduced by the *handshake time* for subsequent queries, e.g., by using `edns-tcp-keepalive` on TCP-based sessions. In addition, the stated *handshake times* can further be optimized *between* sessions by the usage of protocol mechanisms such as `TCP Fast Open (TFO)` using TCP, or `0-RTT` for TLS 1.3 ("early data") using TCP as well as QUIC (see Sect. 5). Hence, we investigate the 1,204 *DoQ-verified* resolvers for the support of `edns-tcp-keepalive` and `TFO`, as well as TLS 1.3 `0-RTT` for QUIC. We do not explicitly investigate the support of TLS 1.3 `0-RTT` for TCP, which we instead leave open for future work. For `edns-tcp-keepalive`, we find support only on the *AdGuard* resolvers, although they respond with a `timeout` value of `0` which instructs the client to directly close the session after having received the response. As for `TFO`, we find 208 resolvers supporting the TCP extension, of which no resolver is included in our *DoX-verified* set. Finally, none of the *DoQ-verified* resolvers offer support for QUIC `0-RTT`. However, the lack of 0-RTT support might be a deliberate choice, as the use of 0-RTT exposes clients to privacy risks [39].

Figure 3 presents the distribution of the *resolve time* and the *RTT* (a, left), as well as the *handshake time* and *handshake-to-RTT ratio* (b, right) of the response time measurements toward the 264 targeted *DoX-verified* resolvers; please note the different x-axis scales. The steps in the CDF lines can be explained by two consecutive hops that have a high difference in their latencies, e.g., when crossing continental borders. For our response time analysis, we only consider measurements which successfully return a valid DNS response containing a `Response Code` within a timeout of 5 s. Further, we limit the analysis of DoQ, DoH, DoT, and DoTCP to measurements for which the corresponding *RTT* measurement is successfully answered by the resolvers. For DoUDP, we exclude the *RTT* measurements, as they were not replied to by any resolver, but include the *resolve* time for comparison. Analyzing the *resolve* time in Fig. 3(a, left), we observe that the distributions of all protocols are almost identical, and match the distributions of the respective *RTT* as shown in the subplot. Hence, we confirm that the *resolve* time indeed resembles 1 *RTT*, regardless of the protocol.

In contrast to *resolve time* and *RTT*, the *handshake time* presented in Fig. 3(b, right) shows a vastly different picture. DoTCP (green line) offers the fastest *handshake* times over all protocols with a median of 156 ms (mean 153 ms), which is expected due to DoTCP only requiring the TCP handshake (1 *RTT*) for the establishment of the session. On the other hand, DoT (gray line) and DoH (magenta line) show almost identical *handshake* times, with medians of around 322 ms and means of around 315 ms. As both protocols require the TCP handshake plus the TLS handshake for session establishment, the *handshake* times should resemble 3 times the measured *handshake* times of DoTCP when TLS 1.2 is used (3 *RTTs* in total), and 2 times when TLS 1.3 is used (2 *RTTs* in total): Analyzing the negotiated TLS versions, we observe that 99.6% of DoT and 96.7% of DoH measurements use TLS 1.3, whereas the remaining ones use TLS 1.2. Analyzing DoQ, we find an unexpected result (solid cyan line): Since QUIC combines the connection and encryption handshakes into 1 *RTT*, DoQ is expected to have the same distribution as DoTCP. However, with a median of 235 ms and a mean of 233 ms, the DoQ *handshake* times observed are higher than expected, having its distribution in between the distributions of DoTCP, and DoT and DoH.

To investigate this, we analyze the distribution of the relative number of *RTTs* which are required by the *handshakes* as shown in Fig. 3(b, right, subplot). We divide each successful *DoX handshake time* measurement by its consecutive *RTT* measurement, thus, showing the distribution of the *handshake-to-RTT ratio* of each measurement pair. For DoTCP (green line), we observe that the *handshake* resembles 1 *RTT* as expected. Moreover, DoT (gray line) and DoH (magenta line) again overlap and converge into a long tail, roughly resembling the expected 2 *RTTs* for TLS 1.3 up until the median. On the other hand, DoQ (solid cyan line) differs drastically from the expected *handshake* of 1 *RTT*: With around 20% of measurements showing an *RTT* of 1, DoQ converges to 2 RTTs at the 60th percentile; hence, roughly 40% of DoQ measurements require more than 2 *RTTs*, which is twice as much as expected in comparison. To investigate this, we analyze the `qlog` [32] outputs recorded during our response time measurements, which enable us to analyze the packet exchanges in detail. Using the `qlogs`, we confirm that the response time measurements are not affected by QUIC's `Version Negotiation` feature, as we use the previously negotiated QUIC `Version` of the cache warming session for the handshake of the subsequent DNS response time measurement session (see Sect. 2). However, we attribute the additional 1 *RTT* to the `Address Validation` feature of QUIC, which is a requirement for every session to prevent traffic amplification attacks by validating that the client is able to receive packets. To perform `Address Validation`, the QUIC standard [43] defines 1 *implicit* and 2 *explicit* mechanisms, with the *implicit* mechanism validating the address by receiving a packet protected with a handshake key (i.e., 1 additional *RTT*). The first *explicit* mechanism uses a `Retry` token sent by a server as a response to the clients Initial frame, instructing the client to re-issue the Initial frame with the server-constructed token (i.e., 1 additional *RTT*). The second *explicit* mechanism also leverages a server-constructed token: If a server issued a token using a New_Token frame in a previous session, it can be used in the Initial frame of a subsequent session (i.e., no additional *RTTs*).

Analyzing the qlogs, we find that in every cache warming session a Retry token is sent, and the client is validated using the first *explicit* mechanism. Moreover, we observe that a New_Token frame is also issued in every cache warming session, which we use in the subsequent DNS response time measurement session in order to validate the address within the clients first Initial frame. For those subsequent measurements, we confirm that every DNS response time measurement session is not affected by an additional Retry frame and, thus, no additional *RTT*, as the Address Validation is fulfilled. However, we find that resolvers still enforce the traffic amplification limit of 3 times the amount of data they received despite successful validation of the client's address: Depending on the X.509 certificate issued by a server, its size might exceed the traffic amplification limit, which requires the client to ACK data before the server sends the remaining bytes. Hence, an additional *RTT* is required, resulting in 2 *RTTs* in total as observed in roughly 40% of DoQ measurements (see Fig. 3, b, right, cyan lines) – 2 times as much as expected.

We further analyze the *handshake times* of cache warming queries: This allows us to investigate the effect of Address Validation mechanisms on the DoQ *handshake time* required for the first session establishment between a client and a resolver (see Fig. 3, b, right, dashed cyan lines). With a median of 468 ms and a mean of 487 ms, the *handshake* times for the first session establishment are roughly doubled in comparison to subsequent sessions. Analyzing the qlogs, we find that the traffic amplification limit is also enforced in the cache warming sessions following successful Address Validation, which can therefore require up to 4 *RTTs* (i.e., Initial, Version Negotiation, Retry, and Amplification Limit) – 4 times as much as expected.

Both our DoQ *handshake time* analyses of cache warming and subsequent queries show that an already validated address is still constrained by the traffic amplification limit until the client sends another frame, which adds 1 *RTT* to the handshake. However, while the QUIC standard states that the traffic amplification limit is to be enforced *until* a client is successfully validated, we argue that our observations are most likely an unintentional effect of the QUIC implementations used by the DoQ resolvers. Hence, we suggest resolvers to not enforce the traffic amplification limit on already validated client addresses to optimize the performance, which results in a reduction by 1 *RTT* during the *handshake*.

5 Limitations and Future Work

We acknowledge that the selected vantage point introduces a location bias for our measurements, in particular for the measured latencies in the response time analysis (Sect. 4). The highly varying geographical distances to the resolvers (whose distribution exhibits further biases, see Sect. 3) inherently affect the delays, especially if multiple round-trips are required. Hence, we plan to address this limitation by performing measurements from distributed vantage points worldwide to obtain a more representative view on DoQ response times around the globe.

Moreover, we acknowledge that public DNS resolvers often leverage IP anycast, about which we could not find any publicly available information for DoQ resolvers of *AdGuard* and *nextDNS*. In addition, by cross-referencing anycast IP addresses of public DNS providers used in related work measuring DoT and DoH [33,49,55], we were not able to identify these public DNS providers within our set of *DoQ-verified* resolvers.

Further, we miss resolvers that do not accept DoQ requests without Server Name Indication (SNI) information. For instance, Google requires queries over TLS 1.3 (which, thus, also affects QUIC) to use the SNI extension [13]. As a result, DoQ queries to 8.8.8.8 are not responded to by Google, whereas queries with the `HostName` set to `dns.google.com` in the SNI extension do trigger a DNS response. Since we do not include SNI in our requests due to not knowing the corresponding `HostName` for every identified resolver, we cannot identify and measure resolvers with such requirements. Therefore, the list of DoQ resolvers measured in our study is not exhaustive, as we only consider open resolvers that do not require SNI. Moreover, we only consider IPv4 resolvers in our study; future work should also consider scanning the IPv6 address space as a complement, e.g., based on IPv6 hitlists [34].

Finally, we plan to further evaluate DoQ by using previously established sessions for subsequent queries, as well as TLS 1.3 `0-RTT` between sessions in a future study: Both mechanisms reduce the overhead introduced by the *handshake time*, which affects application layer protocols that typically require multiple DNS queries in rapid succession.

6 Conclusion

DNS over QUIC promises to improve on the established encrypted DNS protocols by leveraging the QUIC transport protocol. In our study, we detailed a slowly but steadily increasing adoption of DoQ on resolvers worldwide, where the observed week-over-week fluctuations reflect the ongoing development and standardization process with rapidly changing implementations and services. Analyzing the response times of DoQ, we showed that the DoQ *handshake* times fully utilize QUIC's potential in around 20% of measurements. However, roughly 40% of measurements show considerably higher *handshake* times than expected, which traces back to the enforcement of the traffic amplification limit despite successful validation of the client's address. While this shows still unused optimization potential, DoQ already outperforms DoT as well as DoH, making it the best choice for encrypted DNS to date.

In conclusion, our study provided a first look at DNS over QUIC. However, we presented only a glimpse of the potential of DoQ: With the expectation that the upcoming standardization of DoQ will cause a surge in adoption along with optimizations of existing implementations, future studies will reveal whether DoQ will truly become the *"One to Rule them All"*.

Acknowledgements. We thank Luca Schumann and Simon Zelenski for their valuable support, as well as Jan Rüth and the anonymous reviewers for their insightful feedback.

References

1. AdGuard C++ DNS libraries. https://github.com/AdguardTeam/DnsLibs. (Accessed 31 Jan 2022)
2. AdGuard DNS. https://web.archive.org/web/20211011184753/adguard.com/en/adguard-dns/overview.html. (Accessed 31 Jan 2022)
3. AdGuard DNS-over-QUIC. https://adguard.com/en/blog/dns-over-quic.html. (Accessed 31 Jan 2022)
4. AdGuard DNS Proxy. https://github.com/AdguardTeam/dnsproxy. (Accessed 31 Jan 2022)
5. AdGuard Home. https://github.com/AdguardTeam/AdguardHome. (Accessed 31 Jan 2022)
6. AdGuard Home Release 0.107.0. https://github.com/AdguardTeam/AdGuardHome/releases/tag/v0.107.0. (Accessed 31 Jan 2022)
7. aioquic. https://github.com/aiortc/aioquic. (Accessed 31 Jan 2022)
8. CoreDNS fork for AdGuard DNS. https://github.com/AdguardTeam/coredns. (Accessed 31 Jan 2022)
9. DNS Measurements. https://github.com/mgranderath/dns-measurements. (Accessed 31 Jan 2022)
10. dnslookup. https://github.com/ameshkov/dnslookup. (Accessed 31 Jan 2022)
11. DNSPerf. https://github.com/mgranderath/dnsperf. (Accessed 31 Jan 2022)
12. Flamethrower. https://github.com/DNS-OARC/flamethrower/tree/dns-over-quic. (Accessed 31 Jan 2022)
13. Google Public DNS: TLS 1.3 and SNI for IP address URLs. https://developers.google.com/speed/public-dns/docs/secure-transports#tls-sni. (Accessed 31 Jan 2022)
14. IP Geolocation API. https://ip-api.com/. (Accessed 31 Jan 2022)
15. Misc DNS Measurements. https://github.com/mgranderath/misc-dns-measurements. (Accessed 31 Jan 2022)
16. NextDNS CLI Client. https://github.com/nextdns/nextdns. (Accessed 31 Jan 2022)
17. NextDNS Knowledge Base. https://help.nextdns.io/t/x2hmvas/what-is-dns-over tls-dot-dns-over-quic-doq-and-dns-over-https-doh-doh3. (Accessed 31 Jan 2022)
18. One to Rule them All? A First Look at DNS over QUIC. https://github.com/kosekmi/2022-pam-dns-over-quic. (Accessed 31 Jan 2022)
19. quicdog. https://github.com/private-octopus/quicdoq. (Accessed 31 Jan 2022)
20. Refuse queries without RD bit. https://knot-resolver.readthedocs.io/en/stable/modules-refuse_nord.html. (Accessed 31 Jan 2022)
21. RouteDNS. https://github.com/folbricht/routedns. (Accessed 31 Jan 2022)
22. The ZMap project. https://zmap.io/. (Accessed 31 Jan 2022)
23. Verify DoQ. https://github.com/mgranderath/verify-doq. (Accessed 31 Jan 2022)
24. ZMap DoQ. https://github.com/mgranderath/zmap-doq. (Accessed 31 Jan 2022)
25. Domain names - concepts and facilities. RFC 1034 (1987). https://doi.org/10.17487/RFC1034
26. Domain names - implementation and specification. RFC 1035 (1987). https://doi.org/10.17487/RFC1035

27. Bajpai, V., et al.: The dagstuhl beginners guide to reproducibility for experimental networking research. SIGCOMM Comput. Commun. Rev. **49**(1), 24–30 (2019). https://doi.org/10.1145/3314212.3314217
28. Brandt, M., Dai, T., Klein, A., Shulman, H., Waidner, M.: Domain validation++ for MitM-resilient PKI. In: ACM SIGSAC Conference on Computer and Communications Security. CCS '18, pp. 2060–2076. Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243790
29. Deccio, C., Davis, J.: DNS privacy in practice and preparation. In: Conference on Emerging Networking Experiments And Technologies. CoNEXT '19, pp. 138–143. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3359989.3365435
30. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: fast internet-wide scanning and its security applications. In: 22nd USENIX Security Symposium (USENIX Security 13), pp. 605–620. USENIX Association, Washington, D.C. (2013). https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric
31. Fujiwara, K., Vixie, P.A.: Fragmentation avoidance in DNS. Internet-Draft draft-ietf-dnsop-avoid-fragmentation-06, Internet Engineering Task Force (2021). https://datatracker.ietf.org/doc/draft-ietf-dnsop-avoid-fragmentation/. Work in Progress
32. Fujiwara, K., Vixie, P.A.: Main logging schema for qlog. Internet-Draft draft-ietf-quic-qlog-main-schema-01, Internet Engineering Task Force (2021). https://datatracker.ietf.org/doc/draft-ietf-quic-qlog-main-schema/. Work in Progress
33. García, S., Hynek, K., Vekshin, D., Čejka, T., Wasicek, A.: Large scale measurement on the adoption of encrypted DNS (2021). https://arxiv.org/abs/2107.04436
34. Gasser, O., et al.: Clusters in the expanse: understanding and unbiasing IPv6 hitlists. In: Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, 31 October–02 November 2018, pp. 364–378. ACM (2018). https://dl.acm.org/citation.cfm?id=3278564
35. Hardaker, W.: Analyzing and mitigating privacy with the DNS root service. In: NDSS: DNS Privacy Workshop (2018)
36. Herzberg, A., Shulman, H.: Fragmentation considered poisonous, or: one-domain-to-rule-them-all.org. In: Conference on Communications and Network Security (CNS), pp. 224–232. IEEE (2013). https://doi.org/10.1109/CNS.2013.6682711
37. Hoffman, P.E., McManus, P.: DNS queries over HTTPS (DoH). RFC 8484 (2018). https://doi.org/10.17487/RFC8484
38. Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., Hoffman, P.E.: Specification for DNS over Transport Layer Security (TLS). RFC 7858 (2016). https://doi.org/10.17487/RFC7858
39. Huitema, C., Dickinson, S., Mankin, A.: DNS over dedicated QUIC connections. Internet-Draft draft-ietf-dprive-dnsoquic-08, Internet Engineering Task Force (2022). https://datatracker.ietf.org/doc/draft-ietf-dprive-dnsoquic/. Work in Progress
40. ICANN: RSSAC002 Data. https://www.dns.icann.org/rssac/rssac002/. (Accessed 31 Jan 2022)
41. IETF: DNS PRIVate Exchange WG. https://datatracker.ietf.org/wg/dprive/about/. (Accessed 31 Jan 2022)
42. Iyengar, J., Swett, I.: QUIC loss detection and congestion control. RFC 9002 (2021). https://doi.org/10.17487/RFC9002
43. Iyengar, J., Thomson, M.: QUIC: A UDP-based multiplexed and secure transport. RFC 9000 (2021). https://doi.org/10.17487/RFC9000

44. Kim, D.W., Zhang, J.: You are how you query: deriving behavioral fingerprints from DNS traffic. In: Thuraisingham, B., Wang, X.F., Yegneswaran, V. (eds.) SecureComm 2015. LNICST, vol. 164, pp. 348–366. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-28865-9_19
45. Kirchler, M., Herrmann, D., Lindemann, J., Kloft, M.: Tracked without a trace: linking sessions of users by unsupervised learning of patterns in their DNS traffic. In: Freeman, D.M., Mitrokotsa, A., Sinha, A. (eds.) Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2016, Vienna, Austria, 28 October 2016, pp. 23–34. ACM (2016). https://doi.org/10.1145/2996758.2996770
46. Labs, N.: Measuring the effects of DNSSEC deployment on query load (2006). http://www.nlnetlabs.nl/downloads/publications/dnssec/dnssec-effects.pdf. (Accessed 31 Jan 2022)
47. Learmonth, I.R., Grover, G., Knodel, M.: Guidelines for performing safe measurement on the internet. Internet-Draft draft-irtf-pearg-safe-internet-measurement-05, Internet Engineering Task Force (2021). https://datatracker.ietf.org/doc/draft-irtf-pearg-safe-internet-measurement/. Work in Progress
48. Li, J., Ma, X., Li, G., Luo, X., Zhang, J., Li, W., Guan, X.: Can we learn what people are doing from raw DNS queries? In: IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, 16–19 April 2018, pp. 2240–2248. IEEE (2018). https://doi.org/10.1109/INFOCOM.2018.8486210
49. Lu, C., et al.: An End-to-End, large-scale measurement of DNS-over-encryption: how far have we come? In: Internet Measurement Conference. IMC '19, pp. 22–35. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3355369.3355580
50. Moura, G.C.M., Müller, M., Davids, M., Wullink, M., Hesselman, C.: Fragmentation, truncation, and timeouts: are large DNS messages falling to bits? In: Hohlfeld, O., Lutu, A., Levin, D. (eds.) PAM 2021. LNCS, vol. 12671, pp. 460–477. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72582-2_27
51. Randall, A., et al.: Trufflehunter: cache snooping rare domains at large public DNS resolvers. In: Internet Measurement Conference. IMC '20. Association for Computing Machinery (2020). https://doi.org/10.1145/3419394.3423640
52. Reddy, K.T., Wing, D., Patil, P.: DNS over Datagram Transport Layer Security (DTLS). RFC 8094 (2017). https://rfc-editor.org/rfc/rfc8094.txt
53. Rüth, J., Poese, I., Dietzel, C., Hohlfeld, O.: A first look at QUIC in the wild. In: Beverly, R., Smaragdakis, G., Feldmann, A. (eds.) PAM 2018. LNCS, vol. 10771, pp. 255–268. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76481-8_19
54. Thomson, M., Turner, S.: Using TLS to secure QUIC. RFC 9001 (2021). https://doi.org/10.17487/RFC9001
55. Doan, T.V., Tsareva, I., Bajpai, V.: Measuring DNS over TLS from the edge: adoption, reliability, and response times. In: Hohlfeld, O., Lutu, A., Levin, D. (eds.) PAM 2021. LNCS, vol. 12671, pp. 192–209. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72582-2_12
56. Wicinski, T.: DNS privacy considerations. RFC 9076 (2021). https://doi.org/10.17487/RFC9076
57. Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., Somaiya, N.: Connection-oriented DNS to improve privacy and security. In: Symposium on Security and Privacy, pp. 171–186. IEEE (2015). https://doi.org/10.1109/SP.2015.18
58. ZMap: UDP Data Probes. https://github.com/zmap/zmap/blob/master/examples/udp-probes/README. (Accessed 31 Jan 2022)