Performance of DNS over QUIC

Bart Batenburg
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
b.batenburg@student.utwente.nl

ABSTRACT

IP addresses are impossible for humans to remember, especially when the number of websites is gigantic these days. To combat this the Domain Name System (DNS) exists to automatically find the address for a hostname. The old protocol is insecure since it sends all data unencrypted, which allows bad actors to eavesdrop, and by preference without reliable and error-corrected transport protocols. More secure and reliable alternatives such as DNS over TLS and DNS over HTTPS have so far increased processing requirements and latency. DNS over QUIC is a new proposed protocol over the faster QUIC transport layer, that claims to have less impact on latency while still providing the same amount of security as other secure DNS protocols. On the Internet, nothing gets adopted based on a theoretical improvement though, so research is needed to assess the theoretical performance claims of this new protocol. The paper will describe a methodology to test the new protocol against the other secure transport protocols for DNS and the classical insecure version over normal TCP and UDP. It will also analyse the latency of each protocol to 4 different locations around the globe and conclude from that that the DNS over QUIC proposed protocol is faster in some situations and similar to the other protocols in others. Making DNS over QUIC a good option for further specification and implementation.

Keywords

DNS, QUIC, DNS over QUIC, DNS over HTTPS, DNS over TLS, DoQ, DoH, DoT, performance

1. INTRODUCTION

In today's world, the Internet is huge, with about 3.4 billion users in 2016 [27] and growing with 27.000 new users every hour. 2 billion websites exist [7] with many possible applications attached to them. But already since the ancestors of the Internet, there was a problem.

36th Twente Student Conference on IT Febr. 4th, 2022, Enschede, The Netherlands.

Humans are not great at remembering numbers, and computers are not great with processing text (quickly), especially in a network where packets have to flow from source to location without suffering much processing delay as to not keep the user waiting.

Names of hosts and their corresponding address were thus stored in a central registry, and a central list was kept that could be referenced if a new connection was to be made [28].

Maintaining this quickly became slow and unwieldy and thus an automated naming system was needed. The domain name system was thus created, specified in RFC 882 [31] and RFC 883 [32] and were further extended in RFC 1034 [29] and RFC 1035 [30]

In the Domain Name System (DNS) a tree data structure is built up. A domain name is read from the right to the left. So if for example the address for www.wikipedia.org needs to be found, a resolver will first ask a root server where it can find .org, then it will ask the .org name-server where it can find who is responsible for the wikipedia domain. For the wikipedia domain this is ns0.wikimedia.org among others.

The server at ns0.wikimedia.org then answers the final query with the IP address for the www.wikipedia.org website server and a connection can be made. Figure 1 shows the process.

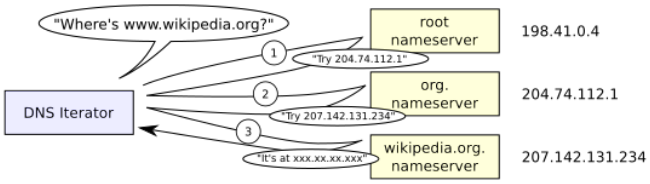


Figure 1: Iterative DNS resolver [24]

DNS has originally been implemented on the UDP transport layer [35], with TCP as a fallback option, sending unencrypted data. This makes it possible for all devices on the route from name server to resolver to see what is being sent and creates the possibility for DNS spoofing and man in the middle attacks because the response packets are not verified or secured in transport. For the spoofing part problem the DNSSEC system was created and specified in RFC 4033, 4034 and 4035 [4, 6, 5]. However, that still leaves the communication readable for everyone on the line. A bad actor that then stores that communication can then possibly use the information for extortion or to specify their attacks against their victim [8].

To fix that last problem, multiple solutions have been thought of. The first to talk about is DNS over TLS, DoT for short, described in RFC 7858 [19] and again further specified in RFC 8310 [12]. This specifies a new extension to the DNS system on a new port, adds encryption and changes the default transport layer to TCP [34].

Another solution is DNS over HTTPS [17], DoH for short. This uses the regular HTTPS connection method [36] and port to access a DNS server. Queries and answers are thus packaged as if they are normal web content. This system is significantly more popular since it uses an already heavily used technology rather than implementing a new DNS extension and can be implemented in browsers while waiting for Operating Systems to implement it [14]. The downside is that DoH contains a full HTTPS stack and thus packet size is larger than directly using TLS. If DoH is implemented in the browser, there is also the possibility of an implementation that uses all the regular web tracking technologies, thus harming privacy again [15].

But the problem with these solutions is the transport layer they run on. TCP [34] is great for making sure that packets arrive at their destination and in order, but that means that if you want to send multiple queries, and the first one suffers a delay or fails, the entire queue has to wait for a re-transmission. This is called head of line blocking. The setup of the connection also takes 4-5 round trips and has to be started again for every connection, where as DNS over UDP is possible in 1 round trip. This all causes a lot of delays [13].

To get rid of the delays, and make the Internet faster, Google designed the QUIC transport layer protocol in 2012. QUIC is built on the UDP [35] transport layer and implements the congestion control, error correction and other elements of TCP in the userspace instead of in the kernel, allowing for quicker updating.

The proposed DNS over QUIC protocol also stops the problem of head of line blocking by multiplexing the connection such that only 1 query is inconvenienced if it happens to suffer an error. In 2021 this protocol was officially standardized into RFC 9000 [22], RFC 9002 [21] and the implementation of TLS1.3 [37] in RFC 9001 [26] to secure QUIC.

Since the QUIC protocol delivers both speed benefits over TCP and security benefits over using UDP, a draft was created for the specifications to run DNS over dedicated QUIC connections, DoQ, to use these benefits for domain resolution [10]. A couple [1] [33] [3] of test setups were also built, with the most talked-about being the recursive resolver run by AdGuard since December 2020.

All the other methods of DNS have been significantly tested [25, 18, 9], but since the DoQ standard is still in the draft phase and has not been adopted in operating systems or regular recursive resolvers and authoritative name-server, to my knowledge no performative study has been done on DNS over QUIC.

To see if DoQ is a viable system, and can help people to access the Internet faster and be safer, its performance needs to be checked and the claims of the draft authors verified. In this paper, we build a setup for testing DNS protocols and we test the performance of DNS over UDP, DNS over TLS, DNS over HTTPS and DNS over QUIC and compare them all to see which protocol delivers the lowest latency.

Besides introducing the research to be performed and delivering some background. This paper further summarizes the problem in section 2 and turns it into a research question in 2.1. In Section 3 we explain the related work found and where it has thus far lacked, leading to reason for this paper. In Section 4 follows the methodology taken in this paper, following with the results of the research in section 5. Lastly, section 6 discusses these results, section 7 concludes this paper, and section 8 lists all the references contained in this paper.

2. RESEARCH CONTRIBUTIONS

Although there are many well designed, tried and tested DNS protocols already researched, to our knowledge no research has been done into the performance of DNS over QUIC until now. This is even mentioned in the draft [10] in chapter 7.1 of some versions. Thus this paper will analyze the performance of DNS over QUIC and conclude if it performs better than the other protocols already in operation.

2.1 Research questions

The research contributions lead to the following questions this research will answer:

Is DNS over QUIC faster than alternatives?

Which can be answered with the following sub-questions:

1. How to build a testing system for DNS protocols?
2. Will DoQ outperform DNS?
3. Will DoQ outperform DoT?
4. Will DoQ outperform DoH/2?

3. RELATED WORK

In this section, a literature review is performed on related work in the field of Domain Name Systems and their performance and in the field of Internet protocols. The Internet has been around for a while already, and some of its roots go back to the start of ARPANET in 1966.

3.1 DNS over UDP

As talked about in the introduction, the Domain name system designed back in 1983 in RFC's [32, 31] and expanded in 1987 in more RFC's [30, 29] is a very well thought of system, but back then the Internet was a tiny place. Not many people were connected and Internet connections were not fast enough for DNS latency to matter. Around 2000 research started to happen on Internet throughput and thought was put into the delay that the DNS system caused [20]. And in 2002 research took place into DNS errors and their effect on Internet traffic [23].

Since then, many large scale studies have taken place on the impact of the Internet expansion towards enormous amounts of users and devices [2], their impact on the DNS system and how to accurately measure it.

Researching impact and performance seems to have become more important since the creation of more advanced Domain Name Systems. This is all because to get systems to implement a protocol there has to be a compelling reason for the developers to put their time into that.

3.2 Encrypted DNS

For DoT itself [19, 12] studies could be found from right after its specification, but after the introduction of DoH [17] studies were conducted in 2019 that analyzed the performance hit of encrypted DNS as a whole [25].

These papers offered inspiration for the possibilities for measuring on a large scale, and also a study of the (then) current situation on the Internet. The setup in these papers was used as inspiration for the testing setup of this paper and the measurements compared were considered for this paper as well.

Another paper about DoH separately [9] has a smaller scale experiment, with local servers and resolvers. The paper measures resolution time and data cost. It found that DNS over HTTPS was slower, but that HTTP/2 based DoH was better equipped to handle Internet delays. The paper however mainly focusses on impact on web traffic, which we will not do. While our research also measures resolution time, it also does so for the new proposed DoQ protocol, which was not researched in these papers.

There are 2 more relevant studies done recently, in 2020 where [18] compares the performance of all the different active DNS protocols in production at that time and [13] conducts a large scale investigation into measuring DoT from the edge with the use of RIPE Atlas. These papers were used as an inspiration for a big measurement system for the various tested protocols, but reading the papers and looking further into the mentioned platforms in them showed there was no support for DoQ at the moment and thus such a test was not possible now.

3.3 DNS over QUIC

Lastly, there is research done on QUIC and the mentioning of the current specification of DoQ. Even though the standardized specifications [22, 21, 26] are only from 2021, performance analysis has been done on the preliminary implementation that Google made of QUIC for itself that was implemented by other big companies as well.

This research can be found in [11] with a good explanation as to where and when QUIC can help. This paper only focusses on web traffic The current specification of DNS over QUIC [10] version 7 will be used as a reference for the theoretical performance improvements the protocol should bring, and that this research will confirm in practise. The draft also contains links to some implementations of the proposed protocol, which can be used for the research methodology.

4. METHODOLOGY

Research comprises of a couple of steps. Firstly, a testing system for the various protocols that were going to be tested was researched and build. Large scale edge measurements on a platform like RIPE Atlas were not possible since access to any platform that could host such a test was not available. And these platforms also had no support for the DNS over QUIC protocol or a way to gain support withing the research period.

In this research we focus on the connection between a recursive resolver and an authoritative name server. The set up is depicted in Figure 2 as well as described in more detail in the text.

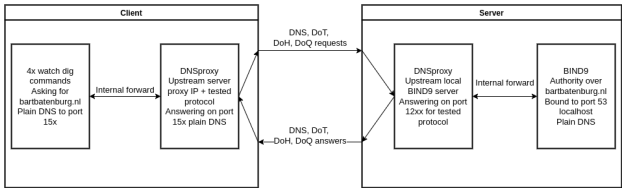


Figure 2: DNS testing server set up

4.1 Authoritative name-server set up

On the server side, BIND9 was selected to act as an authoritative name-server. BIND9 is easy to set it up, is used a lot for this purpose and is the most popular name-server on the Internet [38]. The name-server was setup to be a name server for bartbatenburg.nl. Each server was configured manually to be a name-server for this domain. BIND then answered queries for bartbatenburg.nl with the IP where the website is located.

BIND9 does however not support any other protocols for answering queries than the standard DNS protocol, so a proxy was needed to support all the different protocols that were going to be tested. For this purpose, we chose AdGuard [1] as a proxy.

To the best of our knowledge, Adguard's DNSproxy is the only DNS proxy software that supports all protocols that we want to test in this study. Furthermore, it also exports metrics about the queries it receives and sends to a text file, on which we rely in this study.

4.2 Recursive resolver set up

For consistency, we relied on the same proxy from AdGuard on the recursive resolver side. This simplifies data processing. Other implementations of DNS over QUIC for the client side exist, but lack the features needed for this research.

Since the proxies itself only act as a forwarding layer as you can see in Figure 2, a couple of simple dig commands were run on a loop on the recursive resolver "client" side of the setup. These commands queried the proxy to fetch the A record for bartbatenburg.nl from its upstream, the proxy on the authoritative name-server. A standard DNS query for this A record was 99 bytes, and the answer was 131 bytes. For this loop 4 threads with watch commands were started, each performing a lookup every 2 seconds, for an average of 1 request every 0.5 second.

4.3 Measurements

We focus on the response time between the recursive resolver and the authoritative name server for the remainder of this research. Testing was done over the course of an hour for each protocol. Testing was done to see which protocol has the smallest effect on latency. Other measurements like CPU usage was not tested since the effect was not noticeable.

The proxies still give a fair comparison between the protocols, since the time it takes for BIND9 to respond has been taken out of each measurement. The client side also logs directly from the proxy, so only the communication time between the proxies is measured. This means that all queries should, regardless of its protocol, suffer the same processing delay on the internal forwarding path (see Figure 2). It also gives a realistic image of the performance of DoQ since AdGuard already is using this implementation of DNS over QUIC in production.

4.4 Testing

After this the system as described above was tested locally on 2 virtual machines on the same hardware. Both were configured with 8vCPU's on a Xeon e5 2690v4 and 2GB of RAM. This was for generating a baseline situation where no Internet connection could influence the result, and in order to verify the proxy set up was working as expected.

The test was run for an hour for each protocol (DNS, DoT, DoH and DoQ), storing the logging information for later processing and analysis. After the system was evaluated and deemed working. The server setup was copied to other locations for studying the effects of different levels of network latency on the protocols.

4.5 Authoritative name-server on the cloud

The Microsoft Azure cloud platform was picked for the task of hosting the alternate location servers. For locations the EU West, US East and AU East were picked to have a spread across the globe, and thus different latency levels.

The expectation is that DNS over QUIC will perform better (offer a more stable resolution time) in situations where more latency and longer connections with more possible problems are a factor.

Where the EU data center is in the Netherlands and only 5ms away from the client machine in Enschede, the US East Data center is in the USA and 85ms away and the AU East Data center is in Australia and 248ms away. These different distances are picked to show if there is a difference between the protocols when it comes to handling slower connections.

The machine picked for the task was the B2s server level, with 2vCPU's and 4GB of RAM. More was not needed since the local VM's showed load for the tests was low. All 3 locations were outfitted with one of these servers, and a public IPv4 address was generated by Azure for the connection. After Ubuntu installation, the same BIND9 and DNSProxy setup could then be implemented and testing with each of those servers could start.

4.6 Testing over the Internet

For testing against the servers around the world, the same client machine was used with the upstream address changed to the public IP of the Azure VM, running the same test as before for an hour for each protocol, logging the queries as they flow through the system.

After executing the queries, the generated logs were downloaded and processed. In each log file, all the complete queries and answers were logged, as well as a measurement for how long a query took.

Since not all runs ended up with the same amount of data points because the test was ended early or in the case of Australia each query took significantly longer, some data points were discarded in order to end up with equal measurements. Each test did end up with 6000 data points.

5. RESULTS

In this section, we discuss the performance of DNS over UDP, DNS over TLS, DNS over HTTPS and the new proposed DNS over QUIC in each of the 4 different scenarios as described in the previous sections. The goal of this section is to show and explain the results we achieved and try to conclude from those results what the answer to the research questions are.

Table 1: Mean per location and protocol

	Local (ms)	EU (ms)	US (ms)	AU (ms)
DNS	.55	6.72	84.54	248.99
DoT	.53	7.02	84.88	250.66
DoH	.79	6.70	84.67	248.28
DoQ	.72	6.29	84.19	248.51

Table 2: Variance per location and protocol

	Local	EU	US	AU
DNS	.93	.22	.43	2.43
DoT	.10	9.01	82.14	622.53
DoH	.23	.76	75.32	43.40
DoQ	.10	.31	5.92	43.58

In Figure 3 histograms can be seen for each of the different testing locations. These histograms show the latency of each query and response on the X axis, the Y axis shows the amount of times this latency was experienced and the color shows the protocol that was used for that query. These graphs are printed bigger in the appendix of this paper.

5.1 DoQ against DNS over UDP

Comparing DNS over QUIC to DNS over UDP we see in Table 1 and Figure 3 that locally DNS over UDP is quicker, but when the testing goes over the Internet DoQ becomes quicker. This is surprising as both protocols run over UDP so should behave similarly. DNS over QUIC also requires a connection to be set up and maintained where as DNS over UDP doesn't use connections.

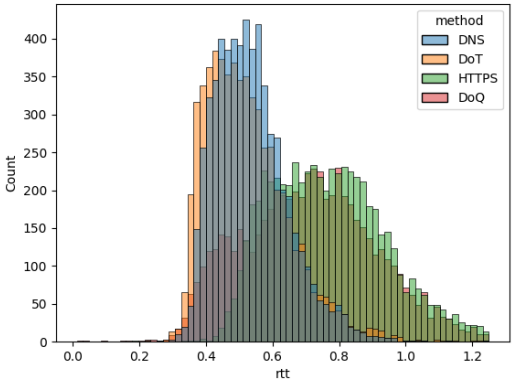
The effect of this setup time can be seen clearly in Table 2 where in the EU, US and AU scenarios that use the Internet the variance of DoQ is clearly higher than that of DNS over UDP.

As to the reason that DoQ is still faster on the Internet than DNS while requiring that setup, we can sadly only theorize. That theory is that because DNS over QUIC keeps the connections open where as DNS over UDP has no connections network equipment handles DoQ differently from DNS over UDP.

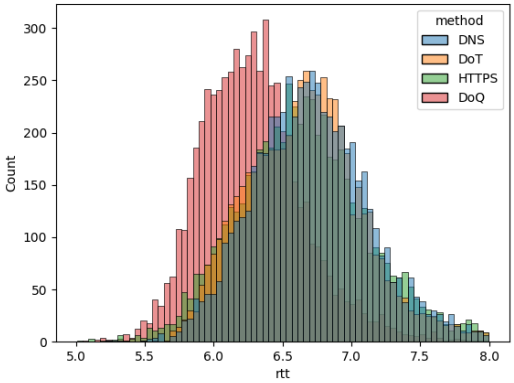
The way that firewalls work is that new connections have to be processed in CPU, where as the established connections are hardware offloaded. This causes the packets for DNS over UDP to have to take the CPU processed route which can be slower. Especially when the firewall is handling a lot of packets like it would in a cloud system like Azure. While the continuous connection of DoQ is already established, and can thus be handled by the faster hardware path.

This theory is also supported by the result of the local set up where there is no network in between the 2 machines that have this effect. Here DNS over UDP can take over in average speed in Table 1 probably because of the lower overhead. The higher variance in Table 2 has no clear explanation.

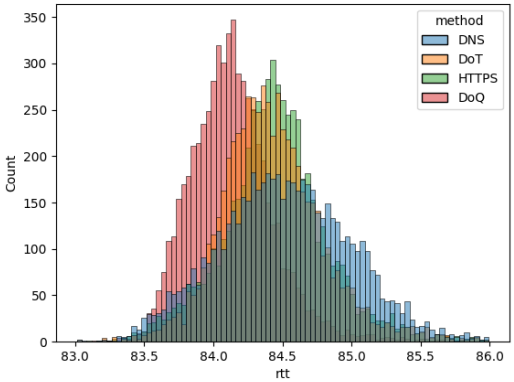
Figure 3: Latency graphs



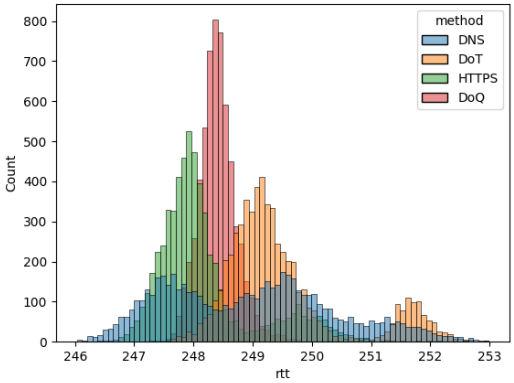
(a) Local



(b) EU



(c) US



(d) AU

5.2 DoQ against DoT

Comparing DNS over QUIC to DNS over TLS there is a clearly superior protocol. DNS over TLS is the slowest protocol in every test over the Internet as can be seen in Table 1. The variance of DNS over TLS is also very high if we look at Table 2.

Both the higher mean and the great variance can be explained by the fact that the DNS over TLS server and client had to redo the set up of the connection a lot during the test, making a significant amount of queries wait more than one round trip time for the answer.

If this was caused by the programming of the proxies is not completely ruled out. But because of the lack of clear effects on all other implementations and the quality with which the proxies seem to have been programmed for the other protocols, this is most likely due to the way the protocol works.

In Australia, a second ”peak” can be seen around the 252ms mark in Figure 3d. This is of interest since it can at least describe some amount of the variance that can be seen in Table 2. The data-points that caused this peak are all close together, thus can be concluded that this effect is caused by some event on the connection to Australia and can be ignored.

5.3 DoQ against HTTPS

Comparing DNS over QUIC to DNS over HTTPS there is a clear winner in the EU and the US, where both the means in Table 1 and variance in Table 2 are lower for the new proposed protocol. Here the lower means are explained by the lower overhead and the more efficient transport layer. The lower variance also means that DoQ is better at maintaining a connection, thus requiring less connection setups after a lost connection is dropped.

In Australia something unexpected happens. Here HTTPS comes out ahead as can clearly be seen in Figure 3d. The lower variance in Table 2 is not represented in the graph however, where it looks that DoQ is more stable.

A reason as to why this happens on the longer distance to Australia from the client in Enschede, was not clear from the data we gathered nor from previously done studies. An explanation can thus not be provided for this.

6. DISCUSSION

The knowledge gathered from the results can be useful in determining if the development and widespread implementation of DNS over QUIC will benefit the Internet. But since the time and resources available for this research where limited, some caveats exist. These elements should be researched before DNS over QUIC is considered to be the best protocol.

Firstly, no test solution was found that could test DNS over HTTPS for HTTP/3, but since the transport layer is the same as DNS over QUIC and the only difference in packets is the size of them, that protocol should receive similar results as DoQ.

Testing with a large amount of clients or a large amount of queries from different locations was also not possible because of the limited time available for research, no access to any big testing network or the money to set one up and the lack of availability of DNS over QUIC implementations in the existing networks.

Since many queries and high resolver/name-server loads were not possible to generate, and because the set up used proxies for communication through the various protocols, any difference in load on the servers was not researched. The effect of DNS over QUIC on processing requirements of recursive resolvers and authoritative name-servers will thus have to be discovered in future work. This is important for implementation because enough processing needs to be installed to handle all DNS requests in a timely manner.

This research also focused on continues connections. This meant that the connection would stay open for as long as the proxies programmed timeout would allow. In the real world however, a recursive resolver would probably not want to keep this connection open for too long, because that takes resources.

The proposed protocol does contain a 0-RTT resumption of the connection which we did not test, so that could be tried to see if it is better compared to the other protocols.

The proxies are also a factor over which there was limited control. The latency between BIND9 and the proxy on the name-server was taken out with the analysis since the proxy logged how much time it took for BIND to respond.

There is a small possibility however, that some programming difference between the protocols exist in the software provided by AdGuard [1]. Future research could look into developing a system without proxies, perhaps when more implementations arrive.

That means that future research should be done into the set up delay of each protocol and the quick connection setup method for previously used connections QUIC has. And other future research could focus on using DNS over QUIC for the connection between a stub resolver and a recursive resolver, where the connection could be kept open for as long as the browser is kept open or maybe even for the entire time the computer is on, since there is a continues need for the recursive resolver anyway.

Research should be done into the reason that DNS over QUIC was faster than DNS over UDP, to see if the established connections of the DoQ protocol really do help in keeping the DNS connections out of the firewall inspections.

And lastly the effect of the new system on web loading times and all-round Internet behaviour would be interesting to know. Research directly into the effect of the implementation of the protocol, from the point of the users of the Internet, will help gaining support of web browsers, public DNS services and Operating System developers.

7. CONCLUSION

In this paper we provide a performance analysis of the new proposed protocol DNS over QUIC, comparing it to other protocols in use in the Internet right now based on the latency of queries.

We show that DNS over QUIC has the advantage in latency over the encrypted protocols in the EU and US set ups, and even performs better than unencrypted DNS over UDP in those situations. Locally it was faster than DoH but lost to DNS and DoT but in Australia DoH took the lead and they both were faster than DoT and DNS.

From this we conclude that DNS over QUIC has a lower effect on latency in short to medium distances to a name-server, while locally and in long distances some other protocols perform a little better.

This paper thus shows that the new proposed protocol is worth developing into a standard, and implementing the standard into systems worldwide. This also seems to be happening, because since this research was started draft 8 of the standard specification was published.

Roll-outs of new protocols are not always easy, looking at the 26 years it has taken for IPv6 to get to 36% adoption [16], but the possibility of implementing DoQ in the user space will help the chances of DoQ significantly. The protocol will then provide the Internet with a quick, secure method of looking up IP addresses.

8. REFERENCES

References

[1] Adguard. *Adguard DNS proxy*. 2021. URL: https://github.com/AdguardTeam/dnsproxy.

[2] Abdelmohsen Ali, Walaa Hamouda, and Murat Uysal. "Next generation M2M cellular networks: challenges and practical considerations". In: *IEEE Communications Magazine* 53.9 (Sept. 2015), pp. 18–24. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7263368.

[3] ameshkov. *amershkov/dnslookup*. Sept. 2020. URL: https://github.com/ameshkov/dnslookup.

[4] R. Arends et al. *DNS Security Introduction and Requirements*. Tech. rep. Mar. 2005. DOI: 10.17487/rfc4033.

[5] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. Tech. rep. Mar. 2005. DOI: 10.17487/rfc4035.

[6] R. Arends et al. *Resource Records for the DNS Security Extensions*. Tech. rep. Mar. 2005. DOI: 10.17487/rfc4034.

[7] Martin Armstrong. *How Many Websites Are There?* Aug. 2021. URL: https://www.statista.com/chart/19058/number-of-websites-online/.

[8] S. Bortzmeyer. *DNS Privacy Considerations*. Tech. rep. Aug. 2015. DOI: 10.17487/RFC7626.

[9] Timm Böttger et al. "An Empirical Study of the Cost of DNS-over-HTTPS". In: *Proceedings of the Internet Measurement Conference*. New York, NY, USA: ACM, Oct. 2019, pp. 15–21. ISBN: 9781450369480. DOI: 10.1145/3355369.3355575.

[10] C. Huitema, S. Dickinson, and A. Mankin. *DNS over Dedicated QUIC Connections*. Tech. rep. URL: https://datatracker.ietf.org/doc/draft-ietf-dprive-dnsoquic/.

[11] Sarah Cook et al. "QUIC: Better for what and for whom?" In: *2017 IEEE International Conference on Communications (ICC)*. IEEE, May 2017, pp. 1–6. ISBN: 978-1-4673-8999-0. DOI: 10.1109/ICC.2017.7997281.

[12] S. Dickinson, D. Gillmor, and T. Reddy. *Usage Profiles for DNS over TLS and DNS over DTLS*. Tech. rep. Mar. 2018. DOI: 10.17487/RFC8310.

[13] Trinh Viet Doan, Irina Tsareva, and Vaibhav Bajpai. "Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times". In: 2021, pp. 192–209. DOI: 10.1007/978-3-030-72582-2{_}12.

[14] Sean Gallagher. *Microsoft says yes to future encrypted DNS requests in Windows*. Nov. 2019. URL: https://arstechnica.com/information-technology/2019/11/microsoft-announces-plans-to-support-encrypted-dns-requests-eventually/.

[15] Glenn Fleishman. *The latest browser privacy feature brings new dangers of its own*. Dec. 2019. URL: https://www.fastcompany.com/90422151/the-latest-browser-privacy-feature-brings-new-dangers-of-its-own.

[16] Google. *IPv6 adoption statistics*. Jan. 2022. URL: https://www.google.com/intl/en/ipv6/statistics.html.

[17] P. Hoffman and P. McManus. *DNS Queries over HTTPS (DoH)*. Tech. rep. Oct. 2018. DOI: 10.17487/RFC8484.

[18] Austin Hounsel et al. "Comparing the Effects of DNS, DoT, and DoH on Web Performance". In: *Proceedings of The Web Conference 2020*. New York, NY, USA: ACM, Apr. 2020, pp. 562–572. ISBN: 9781450370233. DOI: 10.1145/3366423.3380139.

[19] Z. Hu et al. *Specification for DNS over Transport Layer Security (TLS)*. Tech. rep. May 2016. DOI: 10.17487/RFC7858.

[20] Christian Huitema and Sam Weerahandi. "Internet measurements: the rising tide and the DNS snag". In: *Proceedings of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling*. 2000.

[21] J. Iyengar and I. Swett. *QUIC Loss Detection and Congestion Control*. Tech. rep. May 2021. DOI: 10.17487/RFC9002.

[22] J. Iyengar and M. Thomson. *QUIC: A UDP-Based Multiplexed and Secure Transport*. Tech. rep. May 2021. DOI: 10.17487/RFC9000.

[23] Jaeyeon Jung et al. "DNS performance and the effectiveness of caching". In: *IEEE/ACM Transactions on Networking* 10.5 (Oct. 2002), pp. 589–603. ISSN: 1063-6692. DOI: 10.1109/TNET.2002.803905.

[24] Lion Kimbro. *Example of an iterative DNS resolver*. URL: https://commons.wikimedia.org/wiki/File:Example_of_an_iterative_DNS_resolver.svg.

[25] Chaoyi Lu et al. "An End-to-End, Large-Scale Measurement of DNS-over-Encryption". In: *Proceedings of the Internet Measurement Conference*. New York, NY, USA: ACM, Oct. 2019, pp. 22–35. ISBN: 9781450369480. DOI: 10.1145/3355369.3355580.

[26] M. Thomson and S. Turner. *Using TLS to Secure QUIC*. Tech. rep. May 2021. DOI: 10.17487/RFC9001.

[27] Hannah Ritchie Max Roser and Esteban Ortiz-Ospina. "Internet". In: *Our World in Data* (2015). URL: https://ourworldindata.org/internet.

[28] D.L. Mills. *Internet name domains*. Tech. rep. Sept. 1981. DOI: 10.17487/rfc0799.

[29] P.V. Mockapetris. *Domain names - concepts and facilities*. Tech. rep. Nov. 1987. DOI: 10.17487/rfc1034.

[30] P.V. Mockapetris. *Domain names - implementation and specification*. Tech. rep. Nov. 1987. DOI: 10.17487/rfc1035.

[31] P.V. Mockapetris. *Domain names: Concepts and facilities*. Tech. rep. Nov. 1983. DOI: 10.17487/rfc0882.

[32] P.V. Mockapetris. *Domain names: Implementation specification*. Tech. rep. Nov. 1983. DOI: 10.17487/rfc0883.

[33] Nextdns. *Nextdns recursive resolver*. URL: https://nextdns.io/.

[34] J. Postel. *Transmission Control Protocol*. Tech. rep. Sept. 1981. DOI: 10.17487/rfc0793.

[35] J. Postel. *User Datagram Protocol*. Tech. rep. Aug. 1980. DOI: 10.17487/rfc0768.

[36] E. Rescorla. *HTTP Over TLS*. Tech. rep. May 2000. DOI: 10.17487/rfc2818.

[37] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. Tech. rep. Aug. 2018. DOI: 10.17487/RFC8446.

[38] Sean Michael Kerner. *BIND DNS Holds Lead*. Sept. 2020. URL: https://www.serverwatch.com/server-news/bind-dns-holds-lead/.

9. APPENDIX

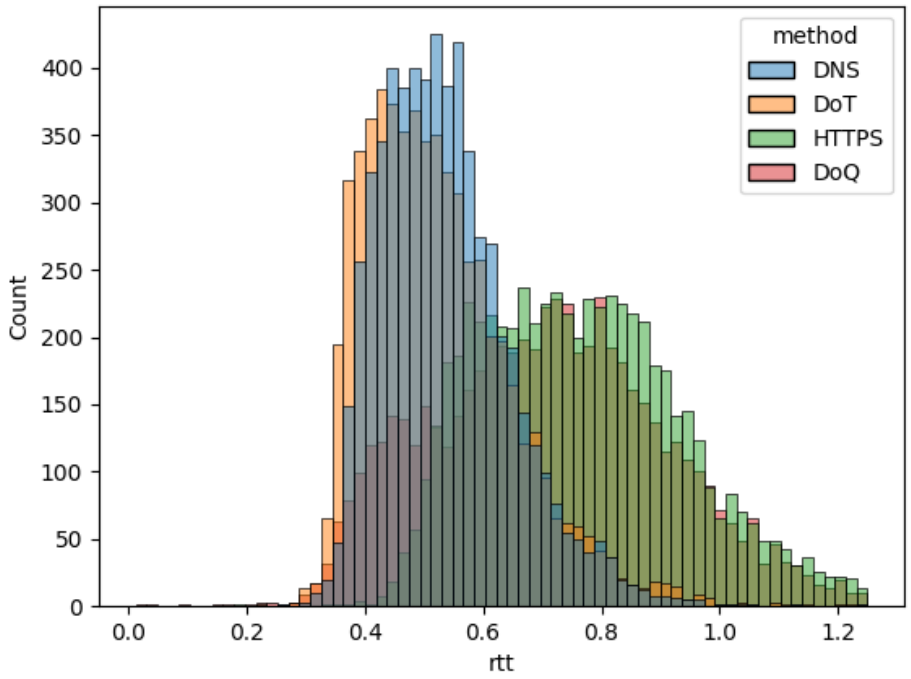


Figure 4: Local

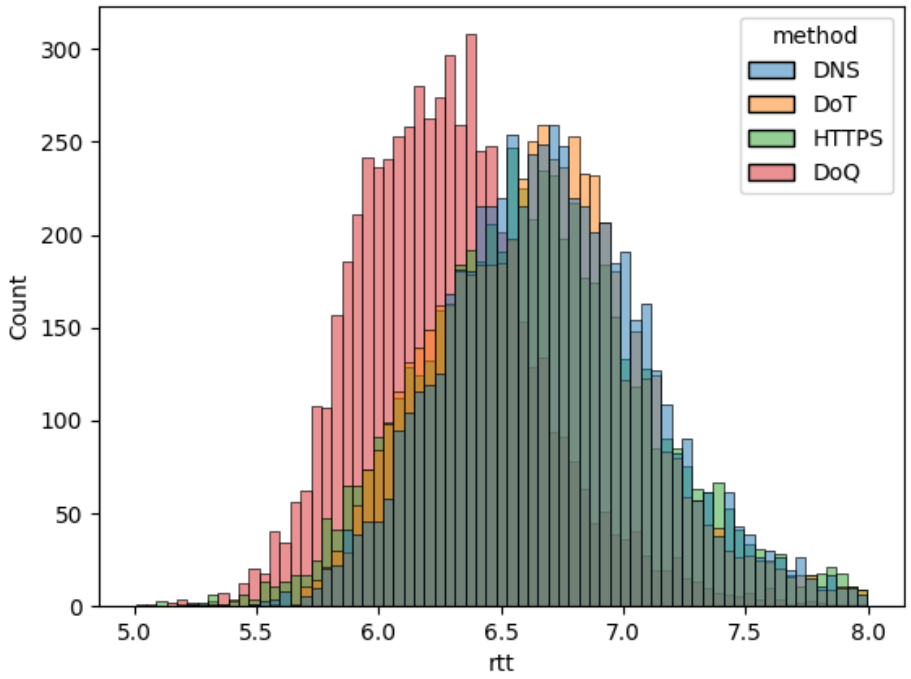


Figure 5: EU

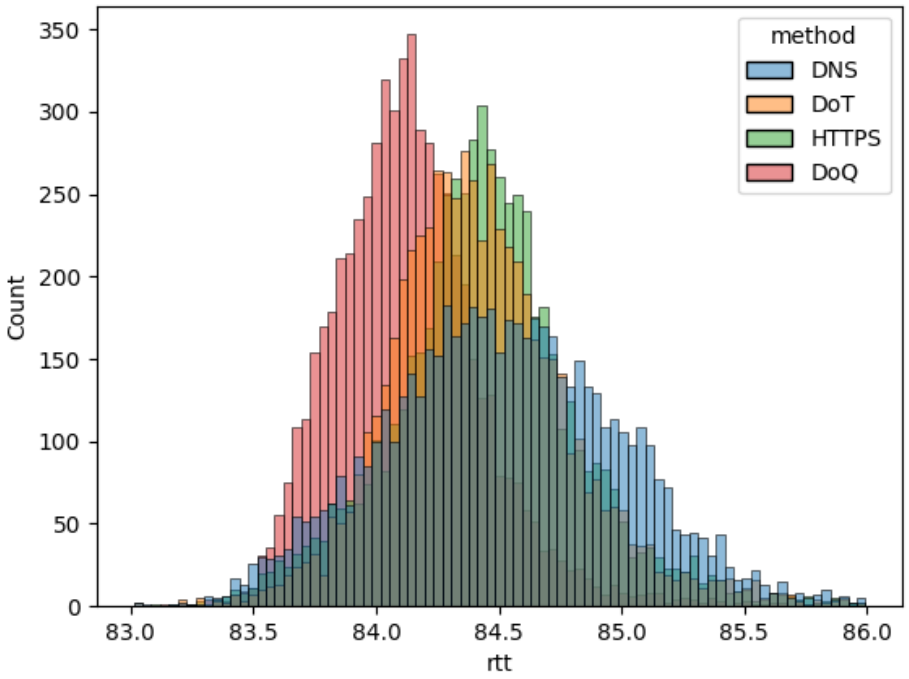


Figure 6: US

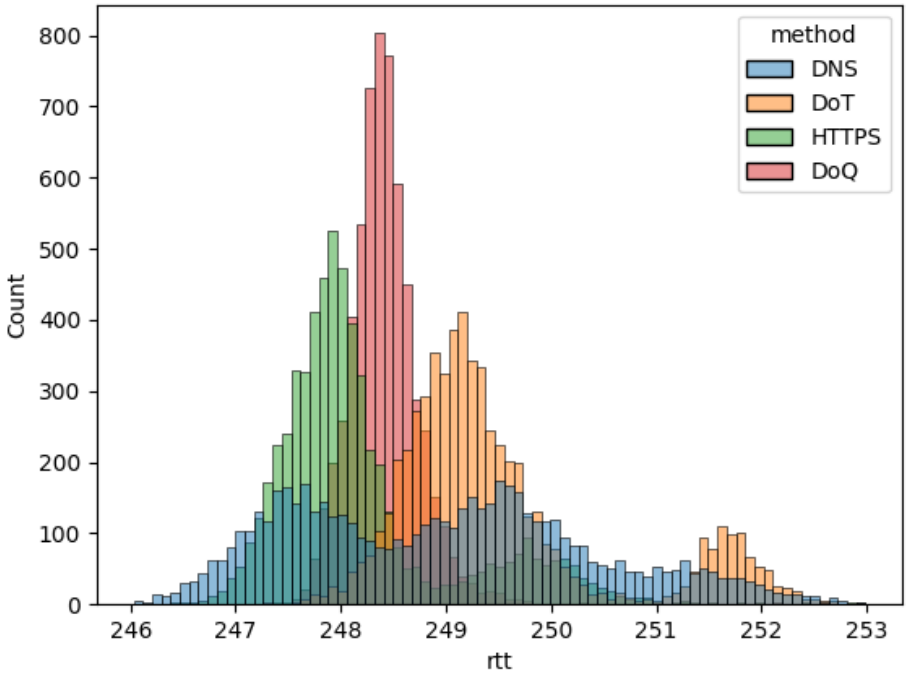


Figure 7: AU