The Effects of DNS Encryption on DNS Resolution and Website Load Time

Brian Y. Li, Henry Gunn High School

Abstract

In this paper, a systematic study was performed to understand the impact of different encryption formats on DNS time and total latency. I tested the top 1000 websites on the Alexa list (*https://www.alexa.com/topsites*) using various DNS encryption schemes: DNSCrypt, DNS over HTTPS, and no encryption, using a Selenium script. The tests produced har files with response time recorded for each website. A parser script was written to extract the relevant metrics from the har files to analyze. I then used the data to do a cost-benefit analysis on using encrypted DNS traffic.

---

1. Introduction

The Domain Name System (DNS) is one of the cornerstones of the modern day internet. It translates hyperlinks and URLs provided by the end user into IP addresses using a series of servers that act in a branching tree formation. As critical as the DNS is to the function of the internet, its design has remained firmly entrenched within the past. Because it was built to function in the most performance-friendly way possible, the design neglected any form of protection for the data it possesses, and this critical shortcoming has never been rectified.

DNS privacy has become an increasing concern among the security world as of late [1], resulting in the emergence of several methods of encrypting DNS traffic. Among them are DNS over HTTP[2] and DNSCrypt[3], which all protect data by carrying it within an encrypted layer.

However, with added security comes increasing performance costs, and due to the relatively new nature of these protocols, there is no native support for these protocols in any operating system. A number of groups, such as Mozilla, are working on developing this technology further, though current explorations are somewhat limited, with only one study detailing the impacts of DNS over TLS and HTTPS on performance as of this writing.

In this paper, I focus on the performance impacts that DNS encryption protocols have on web performance. I believe these measurements are necessary for users to make informed decisions about protocol choice, which allows for a compromise between performance sacrifice and data security. I measure how load times and DNS lookup speeds are influenced by utilising

DNS over HTTPS and DNSCrypt's custom algorithm. I discover that there is a relatively constant value by which DNSCrypt and DoH are slower.

Here, I demonstrate that there is a website loading time increase caused by using DNS encryption, but it is not long enough to detract from the overall web browsing experience. Additionally, I assert that DNS encryption techniques should be studied further and incorporated by the general public. In the following sections, I provide a primer on the functions of the DNS. Then, I detail the high level methodology. After that the detailed description of the experimental setup is provided. Finally I present the experiment results and conclusions.

2. DNS Primer

The Domain Name System was initially designed in 1983 by Paul Mockapetris. Written to provide a way to reference resources across the internet[4], it works using several different layers of servers, each contacting the last. Servers are informed of their search contents by way of a query, which is sent out by the end client then passed on to the other servers. Each query has several components: the header segment, the question segment, the resource record, and a compression section [5]. In the header, the message's overall information, (e.g. ID, status, number of fields, etc.) is contained. The header contains the overall statistics/information about the query/response. The question section contains the QNAME, QTYPE, and QCLASS. The QNAME is the target domain name encoded using a series of labels utilising octet groups, which are units of storage that consist of eight bits each. The QTYPE uses two octet groups to specify the type of query. The QCLASS utilises two more octets to specify the class of the query. The resource records are the basic unit of information and are used for resolving all DNS queries. They come in a variety of different types detailing what information is required. Finally, the compression segment details the domain name, which is compressed into a sequence of labels.

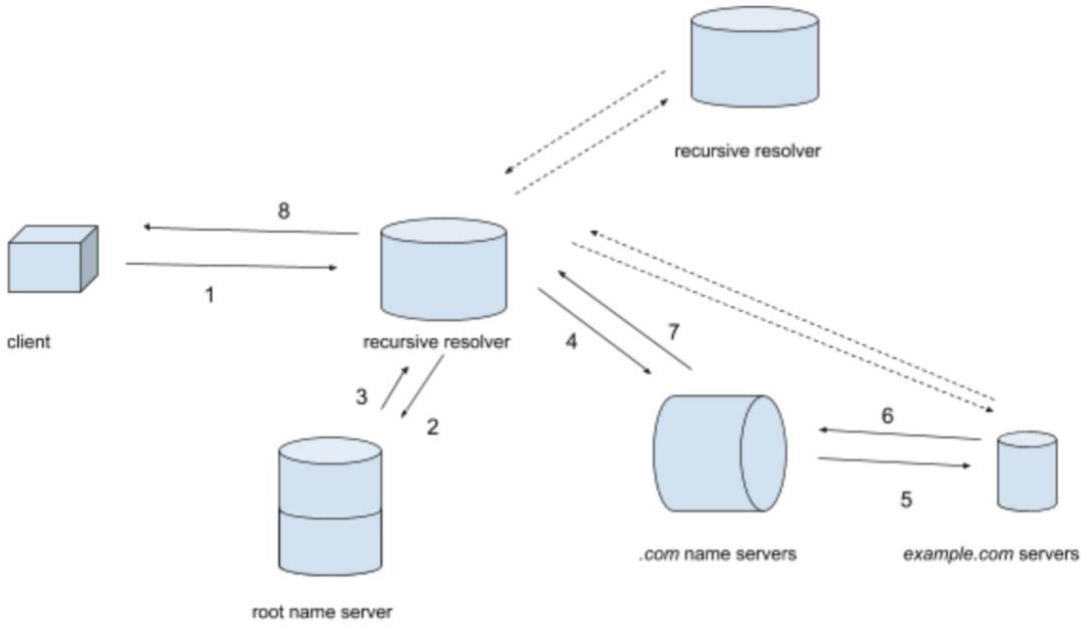

Figure 1: How a DNS query is routed

A DNS query begins with a connection to the recursive resolver or a recursive name server. The recursive resolver receives the query from the client, or end user, and queries a root name server. The root name server acts as a master list of the IP Addresses of all the other servers. In this case, let us assume the client wishes to find the AAAA records of *http://www.example.com*. The client would send a query to the recursive resolver, which would then query the root name server. Locating the address for the desired *.com* servers, the answer would be returned to the recursive resolver. The *.com* servers would be queried by the recursive resolver, which tells the *.com* servers which website specifically they are looking for. In this case, *example.com*. The *.com* servers would query the servers of *example.com*, which would transmit the exact location of the asked-for website. The recursive resolver would receive the answer, and return it to the client.

DNS recursive resolvers will often cache, or store, data, usually the locations of recently queried *.com* servers, to be ready for repeat queries. This helps to vastly speed up response times by simply skipping all the other lookups, directly moving to contact the *.com* servers. For example, if after the previous example query, I wished to find additional records from *example.com*, when the recursive resolver was queried, it would directly contact the *example.com* name servers rather than go through all the other steps.

Sometimes, if a recursive resolver cannot locate the information the user was requesting it might forward the query to a larger recursive resolver with a larger shared cache. That recursive resolver may even forward to an even bigger recursive resolver, so on so forth, explaining the

name recursive resolver. In this manner, the DNS System indexes and stores the IP addresses of all the websites on the internet.

2.1 Vulnerabilities

Given the importance of the DNS, it is surprising how few upgrades the system as a whole has received. As it stands, most of the DNS remains largely the same, with systems retaining much of their original 1983 functionality and design. This means that, most critically for security, the measures implemented upon the system are extremely lacking. The data is almost entirely unencrypted, leading to a host of problems such as phishing and easily stolen data. The data being unencrypted means that just about anyone who has the means can view the traffic being passed to and from the various components of the DNS servers. Even if the user connects using an HTTPS connection, while the normal data will be encrypted under HTTPS, the DNS traffic will not be.

This represents a massive privacy breach. People potentially being granted unrestricted access to one's internet traffic is enormous, considering what kinds of information about one's lifestyle is contained there. From this, one could build up an enormous set of inferences about one's daily lifestyle. In fact, the end users' IP addresses are sent in plain text in the DNS query. This means that, using services like IP geolocation, people can find where the end user lives. In addition, they can gather the data to sell to organizations such as advertising corporations without ones' consent, both from seeing the connection, but also getting the files on the recursive resolver using a program like the Berkeley Internet Name Domain (BIND), the most commonly used DNS software in existence.

2.2 Solutions

As of recent times, solutions have been put into active development by professionals. DNSSEC, the Domain Name System Security Extensions, developed by the Internet Engineering Task Force, was conceived to provide a layer of verification for DNS traffic. It was designed to protect users by adding sets of security keys to DNS traffic. In this manner, DNSSEC prevents people from faking DNS Traffic. However, it does not encrypt the traffic. This role was then filled through several different algorithms, the most prominent of them being DNSCrypt and DNS over HTTPS.

DNS over HTTPS encases DNS traffic in a HTTPS payload to better protect the data. DNSCrypt, developed by an external party, uses elliptic-curve encryption to protect data [6]. These algorithms, however, require both the end user and the DNS server to be actively supporting these methods. This means that if one uses DNS over HTTPS, for example, then they are forced to use DNS servers that also support them. This does not mean that there are few servers supporting both—DNSCrypt is supported by a community among the likes include

iPreadator and Quad9. DNS over HTTPS, on the other hand, is favored by organizations like Google, Mozilla, and Cloudflare. Firefox even has a built-in DNS over HTTPS option.

DNS encryptions have been claimed to be slower because the additional security could increase performance costs and add up over time. However, the effects of these technologies have not been widely studied. In the next section, I discuss our study on the impacts of these technologies.

3. Experiment Methodology

3.1 Metrics

To properly measure the effects of DNS security measures such as DNSCrypt and DNS over HTTPS, I collected the total amount of time spent in DNS lookups, as well as the total website load time, for 1000 websites. The measurements are gathered through HTTP Archive files, or har files. Har files track information that is passed between a website and the computer. In our case, they contain the necessary DNS timings which need to be accumulated to get the total DNS resolution time. Using this information, I look at various statistics that can be gathered through these numbers.

3.2 Experiment Setup

3.2.1 Software

The experiment was conducted using Selenium, using the gecko webdriver to load websites on a 30mb/s connection. Both DNS over HTTPS and DNSCrypt connections were made using a DNScrypt client, called DNSCrypt-proxy. DNSCrypt-proxy gives the user multiple options when it comes to setup. I did not use DNSSEC and connected to Quad9 servers for this. It is important to note here that Feamster et al. have noted that Quad9 performs worse in both standard DNS performance and in DNS over HTTPS [7], Quad9 has been selected by virtue of having the fastest available DNSCrypt supporting servers.

3.2.2 Websites

I collect har files from the top 1000 websites in the Alexa top 1 million list in order to test the effects of using DoH and DNSCrypt on popular, more established websites with reliable connections. In addition, because the top of the list has a lot of diversity, I get to see the effects it has on websites with DNS servers in locations that may already have longer load times using Do53, the current DNS standard. However, in order to prevent websites from stalling out the

data collection, I have included a 60 second timeout timer which prevents websites that load extremely slowly to be excluded. This means that the number of total results per trial may vary.

4. Data and Analysis

4.1 DNS Resolution times

	DNS Median	Percentage Median	Total Median
No Encryption	249.5	6.54%	8153
DoH	344.25	3.40%	12729
DNSCrypt	332	4.32%	8301.5

Table 1: Median Timings for DNS lookups and totals with various encryption formats

Naturally, the most critical aspect I need to look at is the timing of the DNS lookups, since I am trying to determine performance impacts of DNS security. Taking our data, I accumulated the DNS timings using the har files and took the total timings to see what percentage of the total time DNS was taking. Table 1 shows the median DNS time, time percentage, and the median total time aggregated over the 1000 sites. The implications here are fairly clear—I can see that overall just using plain DNS with no encryption is the fastest by almost 100 milliseconds. As a whole, with no encryption, the percentage of the total time used up by DNS lookups is around 2 times higher than the percentage of the total time used up by encrypted DNS connections.

In our specific case, medians are more reliable than the averages, mostly due to many foreign websites (mostly Chinese) generally driving up load times (Table 2). It can be seen that every single statistic is higher when taking averages into account. The most notable change is the drastic increase in total load times. If the average is taken, I can see that each value increases by around 900 or so. What can be noted, however, is that the order of fastest to slowest remains the same. It still takes the longest to load while using DNS over HTTPS, and the shortest without encryption at all. In other words, the pattern still holds, despite using a value that is somewhat skewed.

	DNS Average	Percent Average	Total Average
No Encryption	646.7	9.34%	18336
DoH	770.97	9.34%	20439.23
DNSCrypt	842.35	11.68%	19422.41

Table 2: Average timings for DNS lookups and totals with various encryption formats

I can also see what attributes to these values. Figure 2 displays the DNS Resolution

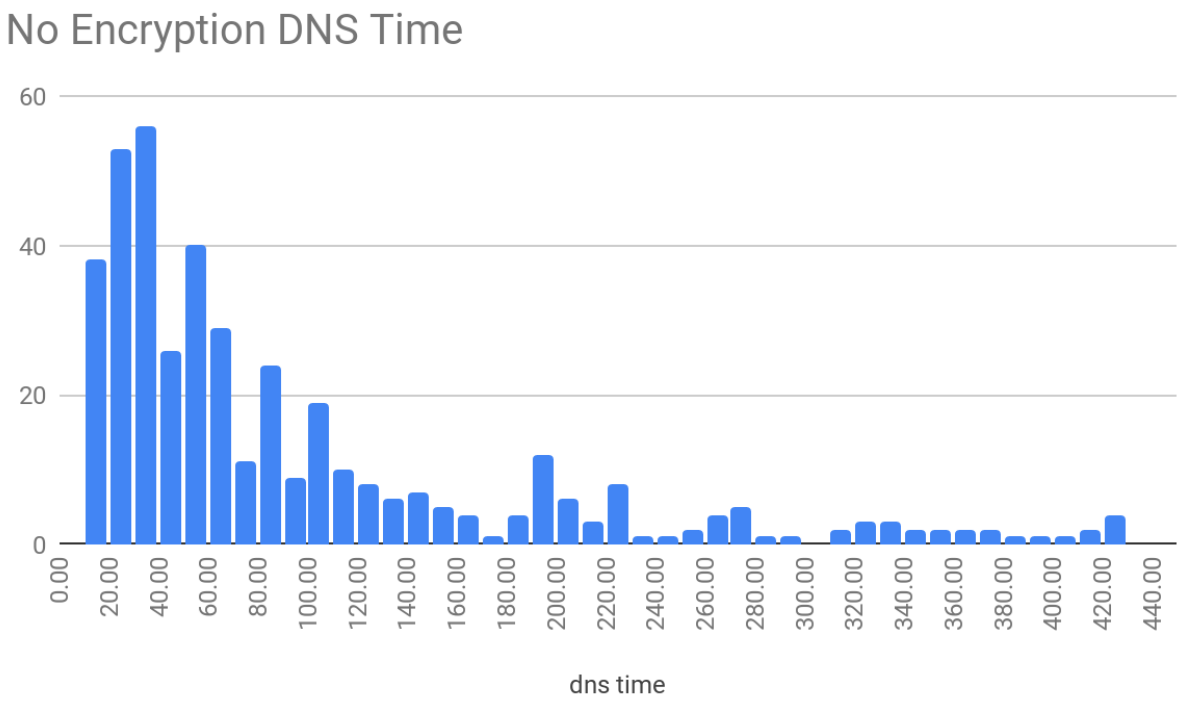

Figure 2: DNS Resolution times for no encryption

times for the entirety of the data. What I can see is that for the most part, the values are centered around 30ms, with a tail end stopping at 440ms. For DNS over HTTPS, it looks fairly similar. Figure 3 displays this. I can immediately see that DoH, unlike default DNS, does not drop off as quickly, and past 90ms, centers around 5 values per 30 millisecond group.

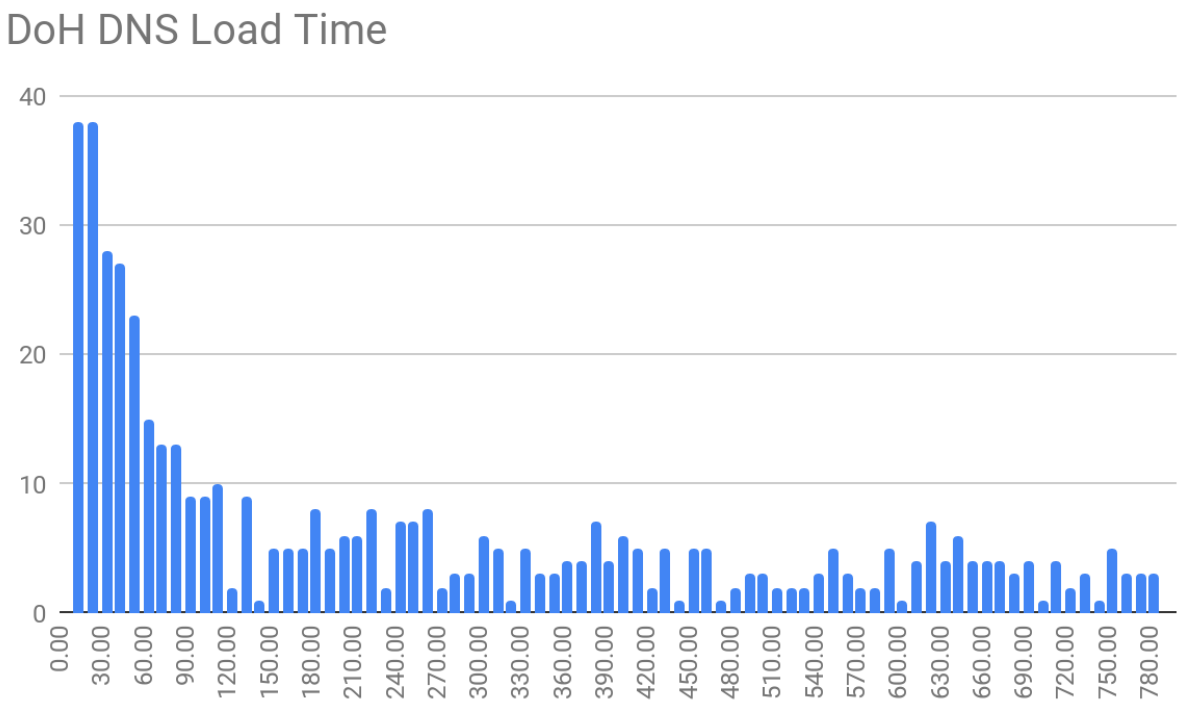

Figure 3: DNS Resolution times for DoH

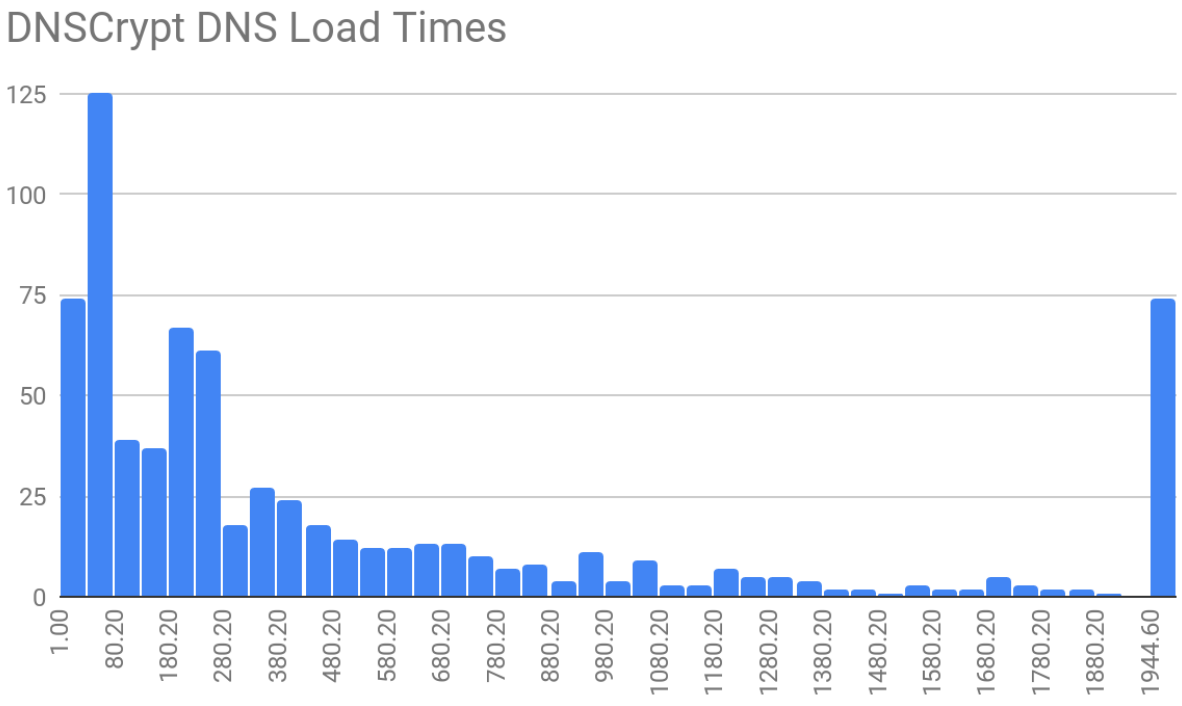

Figure 4: DNS Resolution times for DNSCrypt

	60th Percentile	75th Percentile
No Encryption	361.7	771.25
DoH	503.9	828.25
DNSCrypt	472.3	815

Table 3: 60th and 75th percentile values for the various DNS formats

Finally, Figure 4 displays a histogram for DNSCrypt. Here, I see that while nearly all of the results are above the 60th percentile, given by Table 3, there are 74 values that are greater than 1994.8 and less than 55478.0ms, meaning while DNSCrypt is consistent overall, sometimes it can take much longer than the other formats. The shapes of the graphs also follow expected behavior. I see that all the graphs fall off past 100ms, and no encryption falls off the hardest.

4.2 Page Load times

Because DNS times reflect a specific ordering of speeds, it follows that page load times will have the same ordering. I can see that, both from the average and median of the page

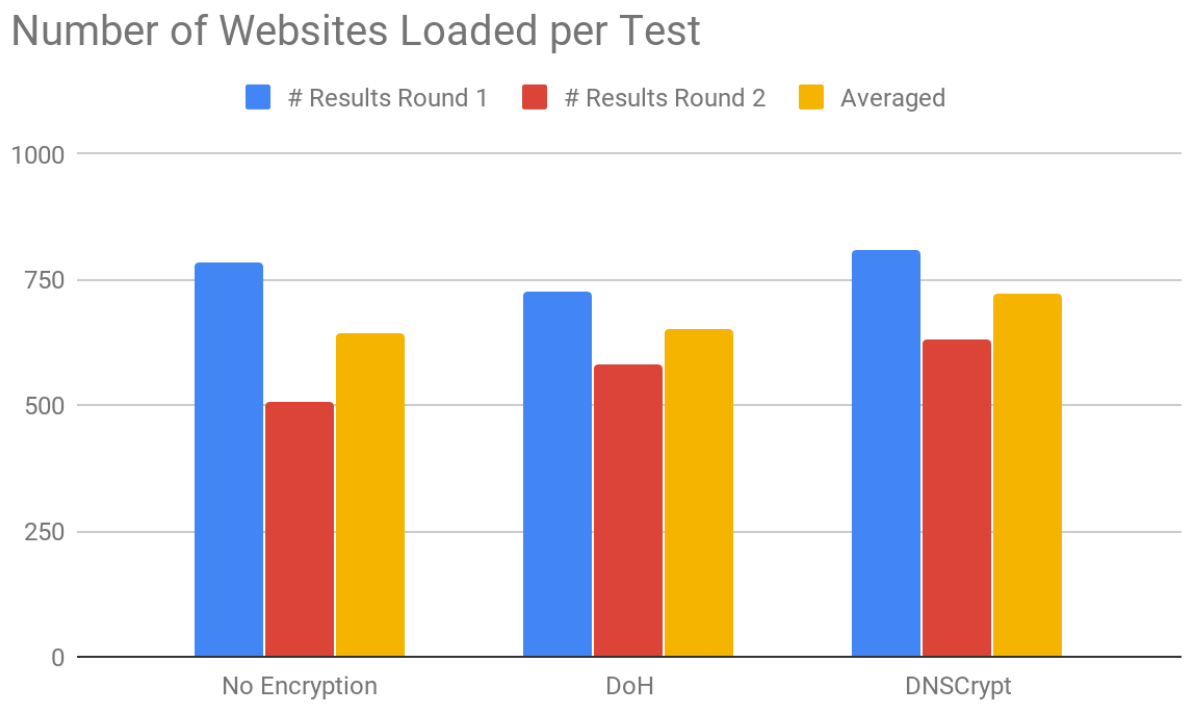

Figure 5: Number of Results for each encryption format

load times, no Encryption still has the fastest load times and DoH having the slowest. However, if I take a look at the number of results (websites not timed out) returned per run, an interesting statistic shows up. On both the first and second trial runs, DNSCrypt loaded more websites above the 30 second mark, while having a lower median and average web page load time. This could indicate that DNSCrypt may perform better than regular encryption when dealing with

slow internet connections or extremely large websites. However, more research is required, because there is very little literature discussing DNSCrypt as of writing.

4.3 Suggestions

While normal DNS is faster than the other protocols, DNSCrypt seems to outperform default DNS in numbers of websites. It also outperformed DoH in both DNS resolution time and website load time. Although DNS over HTTPS is under active development while DNSCrypt is, at the time of writing, several years old, DNSCrypt still is better, especially considering all the support DoH is receiving (google, mozilla, cloudflare, etc.). In addition, DNSCrypt has an active community which runs many servers, providing many more options for users. I suggest that more development be done to integrate DNSCrypt into the Domain Name System because it combines the advantages of DNSSEC with the protection that encryption protocols DoH, for example, provide. It follows that DNSCrypt should also be submitted to the Internet Engineering Task Force (IETF), by way of a Request For Comment (RFC), in order to spread this already polished tool to the public, as well as help support better standards.

5. Conclusion

In this paper, I investigated DNS timings and how they were affected by the use of various DNS transport protocols. I found that overall, while using the DNS protocols I use currently still resulted in fastest resolution and website load speeds, the differences between the protocols were small, varying around 100ms. Additionally, I find that DNSCrypt may outperform DoH in consistently loading webpages, despite being slower in speed.

Based on these results, there are several options for improvement. The protocols behind DNSCrypt should be better examined by the IETF and other organizations to better improve other forms of encryption, such as DoH. In addition, I suggest that DNSCrypt support be integrated into more mainline platforms, such as browsers, and additional DNSCrypt-supporting DNS Servers be set up. Finally, I suggest that the protocols behind DNSCrypt be submitted to the IETF so that better standards regarding DNS Security may be put into place.

6. References

[1] Bortzmeyer. “DNS Privacy Considerations.” RFC 7627 - DNS Privacy Considerations. IETF, August 2015. https://tools.ietf.org/pdf/rfc7626.pdf.

[2] Hoffman, P, and P McManus. “DNS Queries over HTTPS (DoH).” RFC 8484 - DNS Queries over HTTPS (DoH). IETF, July 3, 2018. https://datatracker.ietf.org/doc/rfc8484/.

[3] “Home Page of the DNSCrypt Project [DNS Security].” Home page of the DNSCrypt project [DNS security], n.d. https://dnscrypt.info/.

[4] Mockapetris, P. “DOMAIN NAMES - CONCEPTS and FACILITIES.” RFC 882 - DOMAIN NAMES - CONCEPTS and FACILITIES. IETF, November 1983. https://tools.ietf.org/pdf/rfc882.pdf.

[5] Mockapetris, P. “DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION.” RFC 1035 - DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. IETF, November 1987. https://tools.ietf.org/pdf/rfc1035.pdf.

[6] “DNS Security with DNSCrypt.” OpenDNS. Accessed August 27, 2019. https://www.opendns.com/about/innovations/dnscrypt/.

[7] Hounsel, Austin, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. “Analyzing the Costs (and Benefits) of DNS, DoT, AndDoH for the Modern Web.” 1907.98089.pdf. arXiv, July 18, 2019. https://arxiv.org/pdf/1907.08089.pdf.

6.1 Further Reading

“A Brief History of the Domain Name System (DNS).” WebHostingSearch, n.d. http://www.webhostingsearch.com/articles/history-of-domains-names.php.

Bortzmeyer. “DNS Query Name Minimisation to Improve Privacy.” RFC 7816 - DNS Query Name Minimisation to Improve Privacy. IETF, March 2016. http://buildbot.tools.ietf.org/pdf/rfc7816.pdf.

Mockapetris, P. “DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION.” RFC 883 - DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION. IETF, November 1983. https://tools.ietf.org/pdf/rfc883.pdf.

Postel, J. “Computer Mail Meeting Notes,” February 8, 1982. https://tools.ietf.org/pdf/rfc805.pdf.

Postel, J. “The Domain Names Plan and Schedule,” November 1983. https://tools.ietf.org/pdf/rfc881.pdf.

Zhu, Liang, Zi Hu, John Heidemann, Duane Wessels, Allison Mankin, and Nikita Somaiya. “Connection-Oriented DNS to Improve Privacy and Security.” Connection-Oriented DNS to Improve Privacy and Security. 2015 IEEE Symposium on Security and Privacy, 2015. https://nymity.ch/tor-dns/bibliography/pdf/Zhu2015a.pdf.