Eeny, Meeny, Miny, Moe: Analyzing and Comparing the Selection of DNS Lookup Tools

Jose C. C. Pinto, Eder J. Scheid, Muriel F. Franco, Lisandro Z. Granville

Institute of Informatics (INF), Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil
{jccpinto, ejscheid, mffranco, granville}@inf.ufrgs.br

***Abstract*—The performance of Domain Name System (DNS) resolvers is crucial, as most of the communication on the Internet starts with a DNS lookup to resolve a domain of an IP address to reach the desired content. In this sense, academia has been devoted to measuring and analyzing the performance of DNS resolvers using different tools, either tailored for each work or generic. However, such tools might present different results due to their implementation and affect the measurements. Therefore, this paper reviews the literature on DNS performance research to gather the tools and DNS resolvers most used and, based on this, provides an analysis and comparison of the different DNS lookup tools employed in the literature and discusses the impact of tool selection on measurement results. Research showed that tool selection has an impact on results but not on the lookup success rate.**

***Index Terms*—DNS, Services and Protocols, Measurement**

I. Introduction

Established in 1983, the Domain Name System (DNS) emerged as a critical component of the Internet [1]. Its primary function is to translate user-friendly hostnames (*e.g.*, *wikipedia.org*) into their corresponding Internet Protocol (IP) addresses, effectively serving as the "phone book" of the Internet [2]. Almost all Internet communication starts with a DNS lookup, and complex websites which require content from multiple third parties might perform hundreds of DNS requests before loading a single page [3]. Throughout the history of DNS design and implementation, efforts have been made to minimize latency, such as providing recommendations on how DNS operators can optimize a DNS service to minimize latency for several clients [4]. Thus, DNS performance is of concern, as it directly impacts performance in most Internet-based communications [5].

Past work has extensively measured DNS performance under different conditions. For example, [6] thoroughly analyzed DNS performance with distributed measurements across more than 50 different Internet Service Providers (ISP), in more than 28 countries, comparing local and public DNS resolvers. [7] focused on comparing the performance between DNS and its encrypted versions, DNS-over-TLS (DoT) and DNS-over-HTTPS (DoH), and their impact on web page loading times. [8] measured DoH performance overhead, as well as malicious domain protection. However, they all use different DNS lookup tools (*e.g.*, *dig*, *dnspython*, and *pydig*). When evaluating DNS performance, the lookup tool used to perform the measurements might introduce additional overhead and skew the results, underscoring the necessity for careful tool selection when designing experiments.

In this paper, we compare the performance of different DNS lookup tool libraries, specifically *pydig*, *dnspython*, and the native *dig* Linux command. For that, we review the literature to gather the most widely used tools and select three of them as the focus of our analysis. We collect a dataset of measurements by performing several DNS queries to different public resolvers using the selected tools using two different protocols, the classic DNS-over-Port 53 (Do53), the DoH, and DoT (both DoH and DoT extending previous work [9]). Our analysis focuses on Response Time (RT), which is the time elapsed between issuing the DNS request and receiving a response. Our objective is to determine the impact that tool selection could have on DNS performance measurements.

The remainder of this paper is structured as follows. Section II compares related work. Section III describes the selection of the tools for the experiments and details the methodology used in the measurements. Section IV presents the setup of the experiment and discusses the results. Lastly, Section V summarizes key findings and suggests future work.

II. Related Work

To select the tools (*cf.* Section III-A) to be analyzed in this work, we reviewed the literature on research approaches focused on analyzing the performance of DNS resolvers. Although all efforts focused on measuring DNS performance, they varied in objective and scope.

[8] compares the performance, employing the *pydig* tool, and security aspects of DNS resolvers provided by major Italian ISPs with public resolvers from Google and Cisco (*i.e.*, OpenDNS). Although local resolvers exhibit faster response times, the research finds that their security level matches the public resolvers' level, indicating that users do not need to compromise their security for improved performance of public DNS resolvers.

In a similar study, [6] examines the impact of several DNS resolver responsiveness and cache content on applications such as Content Distribution Networks (CDN). With the use of comprehensive measurements across ISPs, relying on the *dig* Linux tool, the study reveals significant disparities in responses due to CDN location awareness and DNS resolver proximity, uncovering limitations in ISPs' DNS deployments and biases in third-party DNS replies.

TABLE I: Review of Literature on DNS Resolvers Performance Research

Reference	Protocol(s)	Lookup Tool	List of Analyzed Resolvers
[8]	Do53, DoH	pydig	Google, OpenDNS
[6]	Do53	dig	Google, OpenDNS
[7]	Do53, DoH, DoT	dnspython	Google, Cloudflare, Quad9, CleanBrowsing, PowerDNS, BlahDNS, SecureDNS, Rubyfish, Commons Host
[10]	Do53, DoH	dns-measurement	*https://github.com/dnscrypt/dnscrypt-resolvers*
[11]	Do53, DoH, DoT	SamKnows	Anonymized Public Resolvers
[12]	Do53, DoT	RIPE Atlas	Google, Cloudflare, Quad9, CleanBrowsing, UncensoredDNS
[13]	Do53, DoH, DoT	dns-measurement	Google, Cloudflare, Quad9
[14]	Do53, DoH	Firefox	Google, Cloudflare, Quad9
[15]	Do53, DoH	BrightData	Google, Cloudflare, Quad9, NextDNS

[7] investigates two encrypted DNS protocols, DoH and DoT. Using the *dnspython* lookup tool, the authors evaluate the DoH landscape and compare it with the DoT landscape. Furthermore, the study quantifies the impact of DoH on Web page load times, indicating that the protocol provides enhanced security with minimal impact on loading time performance.

In [10], encrypted DNS resolvers that support DoH are evaluated to address privacy concerns. The research shows that while some non-mainstream resolvers have higher response times, there are exceptions, indicating the possibility for users to utilize a broader range of encrypted DNS resolvers than those currently available in popular browser configurations.

[11] investigates the performance of encrypted DNS protocols and conventional DNS in home networks from the United States of America (USA). The research, conducted using a proprietary tool called *SamKnows*, revealed that privacy-focused DNS protocols, such as DoT, could outperform conventional DNS regarding response times for specific resolvers, even with increased latency. The study underscores the need for DNS clients (*e.g.*, browsers) to evaluate latency and response times periodically, suggesting that no single DNS protocol or resolver universally outperforms others for all clients.

Similarly, [12] analyzes DoT adoption and performance, leveraging 3200 Réseaux IP Européens Network Coordination Center (RIPE) Atlas probes in home networks. That research reveals a 23.1% increase in DoT support among open resolvers and a low adoption of local resolvers at 0.4%. Although DoT exhibits higher failure rates and response times, local resolvers achieve response times comparable to public ones despite higher failure rates. Thus, it highlights the complexities and regional disparities in DoT implementation.

[13] explores the impact of the Do53, DoT, and DoH DNS protocols on query response times and page load times from global perspectives using the same tool as [10]. Although DoH and DoT exhibit slightly higher response times than Do53, they can outperform Do53 in terms of page load times. However, in reduced throughput and increased latency conditions, Do53 becomes the fastest option for web page loading. Furthermore, Do53 and DoT show higher success rates in loading web pages compared to DoH. The research suggests strategies for enhancing DNS performance, including opportunistic partial responses and wire format caching, to address varying conditions and improve user experience.

Still in the DoH context, [14] discusses the policy implications of DoH. The authors systematically analyze DoH DNS resolvers, measure DoH's performance effects on Web page loading times using the Firefox Web browser, examine the competitive landscape of such an area, and explore the impact on stakeholders, such as ISPs and consumers. The work sheds light on the potential regulatory and policy implications of widespread DoH deployments.

Lastly, [15] investigates the performance, using the BrightData network, of DoH using a comprehensive dataset from 22 052 clients across 224 countries and territories. That research reveals mixed impacts on the performance of DoH-enabled DNS resolvers. It highlights geographic disparities in DoH and Do53 resolution times, with clients from countries with low investment in Internet infrastructure being more prone to slowdowns when switching to DoH.

In summary, [8] and [6] focused on comparing local and public resolver performance. [7], [10]–[12], [15] investigated the performance impact of using encrypted DNS through HTTPS or TLS protocols, while [13] and [14] do so with additional attention to Web page loading times. Table I presents a detailed review of these efforts and the tools used.

III. Analyzing and Comparing DNS Lookup Tools

This section presents the reasoning behind the selection of the tools based on the literature research conducted, details the methodology employed to compare the tools, and describes the implementation of the approach to perform the queries.

A. DNS Lookup Tools Selection

Regarding DNS lookup tools, we could observe different approaches based on Table I, including the use of proprietary monitoring software such as SamKnows [16] and BrightData [17] but also a non-commercial distributed monitoring tool, called RIPE Atlas [18]. For this work, we selected open source and accessible tools, which are the Python libraries *pydig* and *dnspython*, as well as the native *dig* Linux command. These tools are described in the following paragraphs.

pydig [19] is a Python wrapper library for the *dig* command-line tool. Therefore, it relies on the native *dig* Linux tool, provided by the *bind* package, which allows users to gather detailed information about DNS records, server response times, and domain configurations. In contrast, *dnspython* [20] is a Python library that implements a full-fledged DNS toolkit from scratch. The toolkit can be used for several DNS-related actions, such as queries and nameserver testing. The toolkit implements its communication using sockets to perform queries to DNS resolvers and interact with DNS servers.

B. Methodology and Implementation

Our resolver data set consists of five widely used public resolvers that we also derived from literature research (*cf.* Table I), being Google, Cloudflare, Quad9, CleanBrowsing, and Adguard. For our domain dataset, we selected the most popular domain on the Tranco list [21], retrieved from September 13, 2023, which was *www.google.com*. For each resolver, the lookup of the selected measurements was performed in a loop 500 times so that a significant sample size and a confidence interval of 95% could be collected during the analysis.

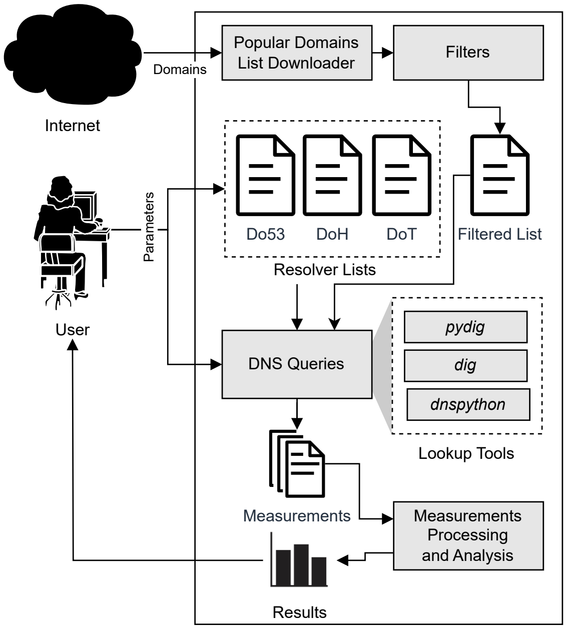


Fig. 1: Measurement Tool Design

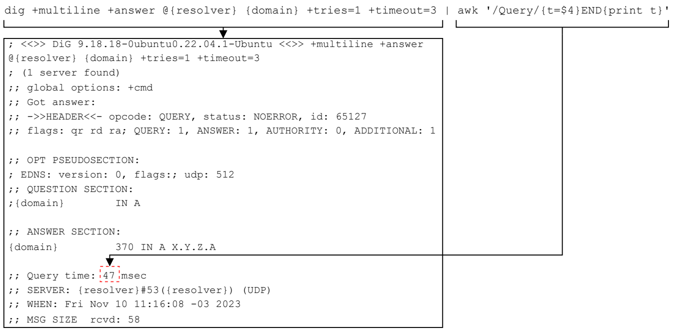
To issue DNS queries from all the tools selected, we designed and implemented a Python measurement tool that executes and collects the results of each combination of tools (*e.g.*, *dig*, *pydig*, and *dnspython*), resolvers (*e.g.*, Google, Quad9, and Adguard) and domains (*e.g.*, *google.com* and *wikipedia.org*). Figure 1 depicts the design of proposed tool including its main components and actors. The tool automatically retrieves the latest list of popular domains curated by the Tranco list project [21], after that, filters (*e.g.*, selection of *n* top domains) are applied in the list to reduce or increase the number of domains to be analyzed. Further, the tool allows the user to input all the parameters, including the resolver lists and the number of DNS requests to be made for each resolver, domain and tool. The tool stores the output of the queries (*e.g.*, return codes, RT, and timestamp) as Comma-Separated Values (CSV) files for post-processing and analysis. This processing is performed in a dedicated component and the results are presented for the user. As the tool was designed to be extensible, other lookup tools can be added by creating a new wrapper to such a tool and including it in DNS queries component.

The performance metric relevant for this study is the RT of a lookup, which consists of the time elapsed between issuing the query and receiving a response from the resolver. To obtain accurate RTs, we measure them using Python's `time` module for each of the tools selected. Listing 1 illustrates the methodology for measuring the RT of a Do53 query using the `time` module. In Line 3, the query is constructed; in Line 4, the start time is recorded to fetch the NS; in Line 5, the query is sent; in Line 6, the end time of the query is captured, and finally, in Line 7, the temporal difference between the end and start times is computed.

```
1 mq = dns.message.make_query(domain, dns.rdatatype.
      NS)
2 start_time = time.time()
3 q = dns.query.udp(mq, resolver, timeout=3)
4 end_time = time.time()
5 res_time = end_time*1000 - start_time*1000
```

Listing 1: RT Measurement of a Request using *dnspython*

To serve as a baseline and to compare with the tool RTs, we also used *awk*, a program that implements the AWK domain-specific language, to extract *dig*'s reported query time as depicted in Figure 2. The figure illustrates the command used to retrieve the *dig* query time in the top of the figure, and, in the solid-line square, the result of a DNS query of a {`domain`} to a {`resolver`} using *dig*. The query time of that specific query is highlighted in the red-dashed square, which is extracted using the *awk* command depicted at the top.



```
dig +multiline +answer @{resolver} {domain} +tries=1 +timeout=3 | awk '/Query/{t=$4}END{print t}'
```

```
; <<>> DiG 9.18.18-0ubuntu0.22.04.1-Ubuntu <<>> +multiline +answer
@{resolver} {domain} +tries=1 +timeout=3
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 65127
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;{domain}        IN A

;; ANSWER SECTION:
{domain}         370 IN A X.Y.Z.A

;; Query time: 47 msec
;; SERVER: {resolver}#53({resolver}) (UDP)
;; WHEN: Fri Nov 10 11:16:08 -03 2023
;; MSG SIZE  rcvd: 58
```

Fig. 2: Extraction of *dig*'s Reported Query Time using *awk*

All source code, domain, and resolver data sets, as well as the measurement results presented and discussed in the next section (*i.e.*, Section IV-A), are available at [22] to promote the reproducibility of the results of the experiments.

IV. Results and Discussion

This section describes the setup of the experiment (*i.e.*, hardware and vantage point) in Section IV-A, presents the results of the Do53 and DoH measurements in Section IV-B and the lookup success rate of both protocols in Section IV-C, and discusses the results and outlines the limitations of our work in in Section IV-D.

A. Experiment Setup

The setup of the experiment was an 2.3 GHz Intel® Core™ i5-6200U machine running Debian 11 Linux operating system, with 16 GB of RAM. To have a stable network connection, the experiments were performed using an 100 Mbps Ethernet cable connected to the Internet with the vantage point being an academic institution located in Porto Alegre (south of Brazil) which has a 10 Gbps dedicated link but no special treatment for DNS-related traffic. Local cache was disabled (*i.e.*, flag `cache` set to `no` in the `/etc/systemd/resolved.conf` file) of the machine used in the measurements to not affect the results.

B. Results

Three different experiments were conducted. The first experiment used the conventional method of sending DNS queries using UDP and port 53 (*i.e.*, Do53). In this experiment, DNS queries were sent without encryption, which potentially leaves the transmitted data vulnerable. In contrast, the second and third experiments used the DoH and DoT protocol, two secure alternatives to Do53 encapsulated within HTTPS or TLS connections, ensuring that the data exchanged between the client and the DNS server is encrypted and secure from interception or tampering by unauthorized parties. By comparing the results of these experiments, it is possible to evaluate the behavior of DNS lookup tools and DNS queries with different protocols, providing insights to analysis using such protocols.

The results of the Do53, DoH, and DoT experiments are shown in Figures 3, 4, and 5, respectively. For each resolver in the *x* axis, different bars are represented, and the *y* axis represents the response time, in milliseconds, measured by each tool. The *dig_awk* bar represents the measurements using the *dig* command in the commmand line to get the DNS query time, *dig_timelib* represents the measurements using the Python's `time` library and calling *dig* from the Python's `subprocess` module, the *dnspython* bar represents the measurements performed using the *dnspython* library and, lastly, the *pydig* bar represents the measurements with the *pydig* library.

1) DNS-over-Port 53 (Do53): Figure 3 shows that the measured RTs of the Do53 queries vary for both different tools and different resolvers. For the same resolver, we can observe RT differences as high as 153% between different tools, as can be seen in the case of *dnspython* (32 ms) and *pydig* (81 ms) mean RTs for the Google resolver. On average, we can see that the best performance comes from the RTs reported of native *dig*, closely followed by *dnspython*. The performance of *pydig* was consistently worse, and similar to that of our implemented *dig* tool measurements using the Python time module.

2) DNS-over-HTTPS (DoH): When employing the DoH protocol to perform queries using the selected tools, a different behavior can be observed, as shown in Figure 4. For example, there was a difference of 1012% between the result of the *dig* reported query time (21 ms) and *dnspython* reported RT (216 ms) for the Google resolver. This behavior is different from the behavior observed using the Do53 protocol, where *dig* and *dnspython* presented similar results. Such a behavior can be explained by *dnspython*'s implementation of the DoH query, which includes the time required to create all the sessions and exchange the keys using HTTPS; whereas the *dig* reported RT only measures the query time.

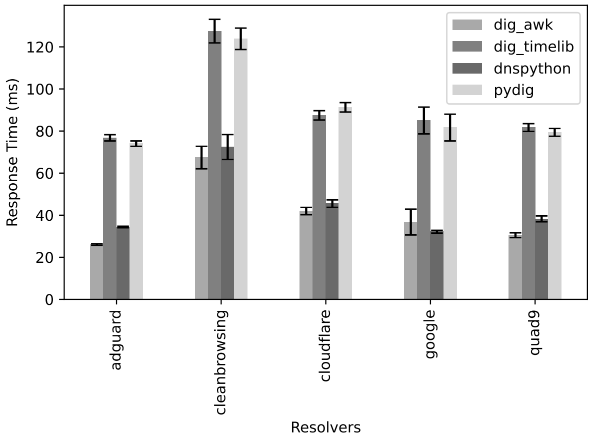


Fig. 3: Results of the Experiments using Do53

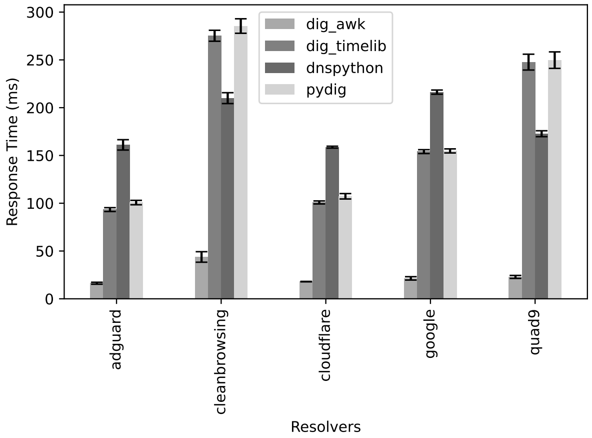


Fig. 4: Results of the Experiments using DoH

3) DNS-over-TLS (DoT): The DoT query results are illustrated in Figure 5. As depicted in the figure, the best performance remains the native *dig* tool. Moreover, the difference between the native *dig* and *dnspython* is still higher on average than the Do53 results (*cf.* Figure 3), which further corroborates the theory that the TLS handshake overhead is not measured by *dig*'s reported RT, only in *dnspython*. Lastly, the performances of *pydig* and *dig* tool measurements using the Python time module are slower on average.

C. Lookup Success Rate

In addition to analyzing the RT of the queries, we analyzed the lookup success rate of all the queries issued for Do53, DoH and DoT. This analysis on the lookup success rate provides a in-depth understanding of the reliability and effectiveness of

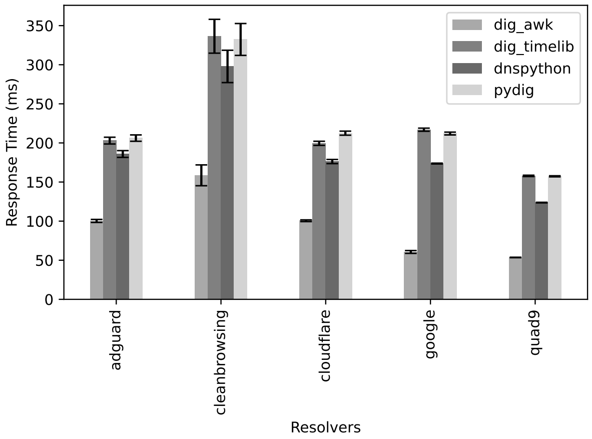


Fig. 5: Results of the Experiments using DoT

the selected tools and selected resolvers in successfully resolving DNS queries. Thus, providing another key contribution of our research and adding another aspect to be considered when selecting the DNS lookup tool.

TABLE II: Lookup Success Rate using Do53

		Lookup Tool			
Resolver	**IP Address**	*dig_awk*	*dig_timelib*	*dnspython*	*pydig*
Adguard	94.140.14.14	99.8%	99.8%	99.8%	100%
Cleanbrowsing	185.228.168.168	93.0%	93.0%	91.6%	91.6%
Cloudflare	1.1.1.1	99.0%	99.0%	98.8%	99.2%
Google	8.8.8.8	92.2%	92.2%	90.2%	92.2%
Quad9	9.9.9.9	92.4%	92.4%	92.0%	94.6%
	Mean Rate	95.3%	95.3%	94.5%	95.5%

Table II summarizes the success rate of the 500 DNS lookup tasks performed for the *www.google.com* domain using Do53 and each of the selected tools on different DNS resolvers. From the mean rate results, it can be seen that there is no significant difference or correlation between the lookup tool and the success rate. However, the success rate is closely related to the DNS resolver that performs the lookup; the rows highlighted in light gray represent the two best DNS resolvers (*i.e.*, the resolvers with the highest lookup success rate). Adguard using the *pydig* tool was the combination able to resolve all lookups without any error (*e.g.*, timeout); Cloudflare was also consistent, in terms of the lookup success rate, for each lookup tool with a 99% mean success rate.

TABLE III: Lookup Success Rate using DoH

		Lookup Tool			
Resolver	**HTTP Endpoint**	*dig_awk*	*dig_timelib*	*dnspython*	*pydig*
Adguard	*https://dns.adguard.com/dns-query*	100%	100%	100%	100%
Cleanbrowsing	*https://doh.cleanbrowsing.org/doh/adult-filter*	100%	99.8%	99.8%	99.8%
Cloudflare	*https://cloudflare-dns.com/dns-query*	100%	100%	100%	100%
Google	*https://dns.google/dns-query*	100%	100%	100%	100%
Quad9	*https://dns.quad9.net/dns-query*	100%	100%	100%	94.6%
	Mean Rate	100%	99.9%	99.9%	99.9%

The lookup success rate of the same queries to *www.google.com* using the same tools and resolvers was collected using the DoH protocol. Table III presents the results of the success rate, which was 100% in all cases, except for the Cleanbrowsing resolver (row highlighted in gray), which failed once during all queries, resulting in a success rate of 99. 8%. Compared to Do53, it can be stated that DoH queries are more stable and will return the result of the query successfully given their use of HTTPS using TCP instead of UDP, which has no delivery guarantee.

TABLE IV: Lookup Success Rate using DoT

		Lookup Tool			
Resolver	**IP Address**	*dig_awk*	*dig_timelib*	*dnspython*	*pydig*
Adguard	94.140.14.14	100%	100%	100%	100%
Cleanbrowsing	185.228.168.168	100%	97.6%	95.8%	97.8%
Cloudflare	1.1.1.1	100%	100%	100%	100%
Google	8.8.8.8	100%	100%	100%	100%
Quad9	9.9.9.9	100%	100%	100%	100%
	Mean Rate	100%	99.5%	99.2%	99.6%

Finally, the same analysis is repeated for queries using the DoT protocol. Table IV presents the results of the success rate, which was 100% in all cases, except for the Cleanbrowsing resolver (row highlighted in gray), which failed due to timeouts (12 to 21 times out of 500, depending on the tool). Similarly to DoH, we can observe higher success rates than when using the Do53 protocol. This behaviour is likely due to TCP feature of DoH and DoT that guarantee delivery, as opposed to UDP which does not offer such a guarantee.

D. Discussion and Limitations

One reason that could explain the performance differences between *dnspython* and *pydig* is the fact that *pydig* acts as a wrapper of the *dig* command, using the `subprocess` module; thus, it requires system calls (*e.g.*, opening a process and reading process descriptors) from Python to the Operating System (OS). In contrast, *dnspython* performs the queries directly in native Python code by relying on native UDP and TCP sockets, which results in faster communication with the resolver compared to *pydig*. Moreover, the *dig_awk* command retrieves the RT of the query directly from the response, not adding the OS overhead in the RT. Thus, it shows that *dnspython* is the closest one to the native *dig* command (*i.e.*, *dig_awk* in Figure 3).

However, this behavior can only be assumed for DNS queries using the Do53 protocol. When performing queries using the DoH and DoT protocol, the *dnspython* tool and *dig* vary considerably, with *dig* presenting the lower RTs of all the investigated tools. Furthermore, *dig_timelib* and *pydig* showed similar results in both Do53 and DoH, which shows that there is no difference when using either. However, *dnspython* presented higher RTs in the majority of resolvers. Thus, the implementation of the DoH protocol in *dnspython* presents an overhead that must be considered for this case. Based on results found, it can be concluded that not all DNS resolvers provide a consistent DNS resolver service for users, which might affect user's browsing and overall Internet usage experience. In addition, such inconsistency was also found in the tools used in the literature to measure the performance of the DNS resolver, leading to skewed results.

A limitation of the work presented herein is the fact that the measurements were conducted from a single vantage point. In this sense, RTs can be affected by network conditions and routing decisions. However, the focus of this work is to analyze the difference in RT between tools and not between DNS resolvers. Therefore, the use of a single vantage point does not invalidate the results presented in this section and the considerations made in this work. In addition, such a limitation is planned to be addressed in future work.

V. Conclusion and Future Work

In this paper, we analyzed the performance impact of using different DNS lookup tools in DNS performance measurements of DNS-over-Port 53 (Do53), DNS-over-HTTPS (DoH), and DNS-over-TLS (DoT) queries. The literature on DNS performance measurement was researched to investigate efforts on such a topic and select which were the most used lookup tools and DNS resolvers in the approaches. On the basis of that, three tools (*i.e.*, *dig*, *pydig*, and *dnspython*) and five public DNS resolvers (*i.e.*, Adguard, Cleanbrowsing, Cloudflare, Google, and Quad9) were selected for our analysis.

From the results of the experiment, we found a significant variation in RT lookups across different tools and resolvers, with performance impacts as high as 153% of Do53 and 1012% for DoH queries. Additionally, there was no correlation between the lookup tool and the DNS lookup success rate, the rate being only related to the DNS resolver used. Further, DoH showed to be consistently more stable in the success rate of the queries in all of the selected tools and resolvers compared to the success rate of queries issued using Do53.

However, based on our findings, we conclude that tool selection directly impacts results when analyzing DNS performance. This difference in results can be explained due to the tool's implementation, which varies from using the Operating System (OS) to call an external DNS lookup tool (*e.g.*, *dig*) or using native Python sockets to create DNS requests (*e.g.*, *dnspython*). Thus, it is suggested that researchers carefully select the tool when designing future experiments and take into account that the results might be affected not because of network conditions or the implementation or deployment of the DNS server but because of the tool they are relying on. Further, this aspect also impacts on the data quality that is used as input for analysis and statistics.

Future work includes, but it is not limited to, *(i)* increase the selection of tools being compared, *(ii)* include tools implemented in different languages, *(iii)* analyze DNS network packets in-depth to gather more granular temporal information, and *(iv)* add diversity of vantage points and network conditions (*e.g.*, mobile networks) of the measurements.

Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP) under the grant number 2020/05152-7, the PROFISSA project.

References

[1] P. Mockapetris and K. J. Dunlap, "Development of the Domain Name System," in *Symposium Proceedings on Communications Architectures and Protocols (SIGCOMM 1988)*, Stanford, California, USA, August 1988, p. 123–133.

[2] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Pearson, 2016.

[3] M. Butkiewicz, H. V. Madhyastha, and V. Sekar, "Understanding Website Complexity: Measurements, Metrics, and Implications," in *ACM Conference on Internet Measurement Conference (IMC 2011)*, Berlin, Germany, 2011, p. 313–328.

[4] M. Müller, G. C. M. Moura, R. de O. Schmidt, and J. Heidemann, "Recursives in the Wild: Engineering Authoritative DNS Servers," in *ACM Internet Measurement Conference (IMC 2017)*, London, United Kingdom, 2017, pp. 489—-495.

[5] I. Bozkurt, A. Aguirre, B. Chandrasekaran, P. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Why Is the Internet so Slow?!" in *International Conference on Passive and Active Network Measurement (PAM 2017)*, Sydney, Australia, 02 2017, pp. 173–187.

[6] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig, "Comparing DNS Resolvers in the Wild," in *ACM SIGCOMM Conference on Internet Measurement (IMC 2010)*, Melbourne, Australia, 2010, p. 15–21.

[7] T. Böttger, F. Cuadrado, G. Antichi, E. L. a. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An Empirical Study of the Cost of DNS-over-HTTPS," in *ACM Internet Measurement Conference (IMC 2019)*, Amsterdam, Netherlands, October 2019, pp. 15—21.

[8] A. Affinito, A. Botta, and G. Ventre, "Local and Public DNS Resolvers: Do You Trade Off Performance Against Security?" in *IFIP Networking Conference (IFIP Networking 2022)*, Catania, Italy, June 2022, pp. 1–9.

[9] J. C. C. Pinto, E. J. Scheid, M. F. Franco, and L. Z. Granville, "Analyzing and Comparing DNS Lookup Tools in Python," in *XX Escola Regional de Redes de Computadores (ERRC 2023)*, Porto Alegre, RS, Brazil, Ocotber 2023, pp. 49–54.

[10] R. Sharma, N. Feamster, and A. Hounsel, "Measuring the Availability and Response Times of Public Encrypted DNS Resolvers," 2022, arXiv 2208.04999, cs.CR.

[11] A. Hounsel, P. Schmitt, K. Borgolte, and N. Feamster, "Can Encrypted DNS Be Fast?" in *Passive and Active Measurement*. Cham: Springer International Publishing, 2021, pp. 444–459.

[12] T. V. Doan, I. Tsareva, and V. Bajpai, "Measuring DNS over TLS from the Edge: Adoption, Reliability, and Response Times," in *Passive and Active Measurement*, O. Hohlfeld, A. Lutu, and D. Levin, Eds. Cham: Springer International Publishing, 2021, pp. 192–209.

[13] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Comparing the Effects of DNS, DoT, and DoH on Web Performance," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 562–572. [Online]. Available: https://doi.org/10.1145/3366423.3380139

[14] K. Borgolte, T. Chattopadhyay, N. Feamster, M. Kshirsagar, J. Holland, A. Hounsel, and P. Schmitt, "How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem," *SSRN Electronic Journal*, 01 2019.

[15] R. Chhabra, P. Murley, D. Kumar, M. Bailey, and G. Wang, "Measuring DNS-over-HTTPS Performance around the World," in *ACM Internet Measurement Conference (IMC 2021)*, Virtual Event, 2021, p. 351–365. [Online]. Available: https://doi.org/10.1145/3487552.3487849

[16] Cisco, "SamKnows - Internet Performance Monitoring," 2023, https://www.samknows.com/.

[17] Bright Data Ltd., "Bright Data - The World's #1 Web Data Platform," 2023, https://brightdata.com/.

[18] Réseaux IP Européens Network Coordination Centre RIPE NCC, "RIPE Atlas," 2023, https://atlas.ripe.net/.

[19] L. Smith, "pydig - Github Repository," 2021, https://github.com/leonsmith/pydig.

[20] Dnspython Contributors, "dnspython Python Library," 2020, https://www.dnspython.org/.

[21] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *Network and Distributed System Security Symposium (NDSS 2019)*, San Diego, USA, Feb. 2019.

[22] J. C. C. Pinto, "Encrypted DNS Benchmark," October 2023, https://github.com/jchagastelles/encrypted-dns-benchmark.

All links were visited in February 2024.