Produce a technical plan for an "Irish" Trusted Recursive Resolver

Tung-te Lin M.Sc.

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Networked System)

Supervisor: Stephen Farrell

11 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Tung-te Lin

November 1, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Tung-te Lin

November 1, 2020

Acknowledgments

...ACKNOWLEDGMENTS...

Tung-te Lin

University of Dublin, Trinity College
11 2020

Produce a technical plan for an "Irish" Trusted Recursive Resolver

Tung-te Lin, Master of Science in Computer Science
University of Dublin, Trinity College, 2020

Supervisor: Stephen Farrell

...ABSTRACT...

Summary

...SUMMARY...

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Trusted Recursive Resolver(TRR) [9] is the plan of the browser company Mozilla to provide better privacy compare with using traditional Domain Name Servers(DNS) [10]. This paper is going to analyze how a TRR plan could be applied in the Republic of Ireland from a technical view.

In Chapter 2, the state of the art , it describes how a DNS server works, the problem of privacy would have in using DNS servers. Next, it describes the solutions for dealing with the problem of privacy so far, including Domain Name System Security Extensions(DNSSEC) [11], DNSCrypt [12], DNS over TLS(DOT) [13], DNS over HTTPS(DOH) [14]. Moreover, TRR is introduced here, it explains why Mozilla created TRR, what is the relation between TRR and DOH, and why this study intends to discuss about the TRR program.

In Chapter 3, it discusses the DNS traffic in Ireland. If people plan to deploy DNS servers to implement TRR in Ireland, they need to know how much DNS traffic would have in Ireland. Thus, this chapter provides the method to estimate the DNS traffic in Ireland.

In Chapter 4, the content is the required software. It discuss that if install a DOH DNS server, What kinds of software do a DNS provider needs? What applications exist in the market for each kind so far? Furthermore, this chapter makes a comparison among those applications, and provide some information from previous studies.

In Chapter 5, the study designed an experiment to understand the performance of a DNS server can possess. The software applications used here are the applications mentioned in Chapter 4. This chapter describes the processes in the experiment, and explains the variables and results of the tests in the experiment.

In Chapter 6, the overall design, which is the last chapter, is going to use the estimated DNS traffic in Chapter 3 and the evaluated performance of a DNS server in Chapter 5 to design a technical plan which can resolve the network traffic of the national scale in the Republic of Ireland.

Chapter 2

The state of the art

2.1 The introduction of Domain Name Server

Domain Name Server(DNS) [10] is the server which converts the domain name to Internet Protocol(IP) address. When users type the domain name on the browser, for example, type ”www.dcard.tw” on Google Chrome, then Google Chrome will send the domain name ”www.dcard.tw” to a DNS server. After that, the DNS server responses the IP address of the domain name ”104.16.204.58” to Google Chrome. Finally, Google Chrome is enable to connet to the server of ”www.dcard.tw” by using the its IP ”104.16.204.58”, because machines need the IP address to find out the location of another machine [15] [16], not domain name [17].

Moreover, there are 2 kinds of DNS servers, which are recursive DNS resolver and authoritative DNS server [18] [18]. Recursive DNS resolvers do not save all the IP addresses and domain names that users need. If users query the domain names that the recursive DNS resolver does not know, then the recursive DNS resolver will inquire authoritative DNS servers about the IP address for the domain name. Next, the recursive resolver saves the domain name and its IP address in the cache, in case any users use this domain name in a short time, then the recursive resolver is able to reply its IP address immediately and not to inquire authoritative DNS servers again.

About authoritative DNS servers, they store IP addresses and domain names that users need, but users can not inquire them directly, because users have to utilize recursive DNS resolvers to inquire authoritative DNS servers to get correspond IP addresses.

Furthermore, authoritative DNS servers are hierarchical [5]. The highest one is root server. A topper level authoritative DNS server can respond a address of a lower level authoritative DNS server to users for their inquiry, and that lower level authoritative DNS server may have the IP address what users need. In that case, a authoritative DNS server does not need to save entire IP addresses and domain names, the workload can be divided. The levels of authoritative DNS servers are shown in Fig. 2.1.

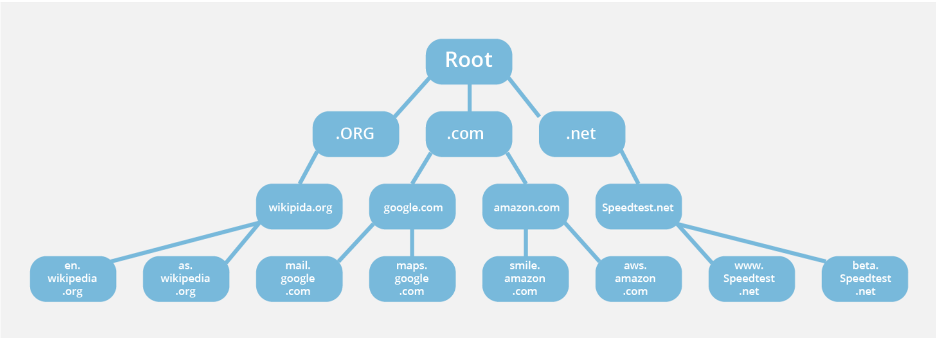


Figure 2.1: *The levels of authoritative DNS servers [5]*

2.2 The problem of privacy

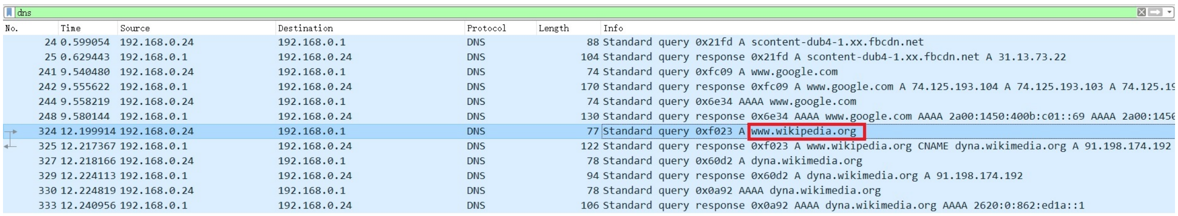
Request For Comments(RFC) [19] is the publication which is managed by Internet Engineering Task Force(IETF). IETF is the organization to make a standard to Internet [20].

In the documents of RFC, RFC7626 [21] and its draft [22], they discussed about the problem of privacy in using a DNS server. The personal data could be discovered in the DNS servers, wire or DNS requests, including the IP of clients, the domain names which are researched by users, even the applications that users use.

The reason to cause the privacy problem is that the typical DNS traffic is not encrypted. On the other hand, DNS servers, especially recursive resolvers, they store the data of DNS queries in their log or cache, thus the DNS providers are able to use the data to do some analysis or transfer the data to others. As for authoritative DNS servers, the privacy problem is lighter than recursive resolvers, because their cache is too limited to store the completed data, but, authoritative DNS servers still have chances receive the personal data such as users' IPs [21].

The typical DNS traffic is not encrypted. Which means, if others get the packets from users, then they may understand what websites users browse or what applications users use from the packets of traditional DNS queries.

In order to understand the situation clearly, this study did a test to get some traditional DNS packets from a user by using Wireshark. Wireshark is the software application for catching packets in a machine. When the researcher typed a domain name of a website on a web browser, then that domain name was displayed on Wireshark. The screenshot is shown in Fig. 2.2.


dns

No.	Time	Source	Destination	Protocol	Length	Info
24	0.599054	192.168.0.24	192.168.0.1	DNS	88	Standard query 0x21fd A scontent-dub4-1.xx.fbcdn.net
25	0.629443	192.168.0.1	192.168.0.24	DNS	104	Standard query response 0x21fd A scontent-dub4-1.xx.fbcdn.net A 31.13.73.22
241	9.540480	192.168.0.24	192.168.0.1	DNS	74	Standard query 0xfc09 A www.google.com
242	9.555622	192.168.0.1	192.168.0.24	DNS	170	Standard query response 0xfc09 A www.google.com A 74.125.193.104 A 74.125.193.103 A 74.125.19
244	9.558219	192.168.0.24	192.168.0.1	DNS	74	Standard query 0x6e34 AAAA www.google.com
248	9.580144	192.168.0.1	192.168.0.24	DNS	130	Standard query response 0x6e34 AAAA www.google.com AAAA 2a00:1450:400b:c01::69 AAAA 2a00:145
324	12.199914	192.168.0.24	192.168.0.1	DNS	77	Standard query 0xf023 A www.wikipedia.org
325	12.217367	192.168.0.1	192.168.0.24	DNS	122	Standard query response 0xf023 A www.wikipedia.org CNAME dyna.wikimedia.org A 91.198.174.192
327	12.218166	192.168.0.24	192.168.0.1	DNS	78	Standard query 0x60d2 A dyna.wikimedia.org
329	12.224113	192.168.0.1	192.168.0.24	DNS	94	Standard query response 0x60d2 A dyna.wikimedia.org A 91.198.174.192
330	12.224819	192.168.0.24	192.168.0.1	DNS	78	Standard query 0x0a92 AAAA dyna.wikimedia.org
333	12.240956	192.168.0.1	192.168.0.24	DNS	106	Standard query response 0x0a92 AAAA dyna.wikimedia.org AAAA 2620:0:862:ed1a::1

Figure 2.2: *The queried website is revealed in packets if using the typical method to query website*

2.3 The solutions for privacy

According to the description of RFC7626 [21], revealing the content of DNS queries can cause the severe privacy problem. People may not be willing to let others to find what websites they browse. Thus, some solutions were created.

There are 4 popular solutions for encrypting DNS queries and they are widely used in different public DNS servers, which are Domain Name System Security Extensions(DNSSEC), DNS over TLS(DOT), DNS over HTTPS(DOH) and DNSCrypt [23].

The first solution is DNSSEC, it is not only the oldest but also the most popular solution among those 4 solutions [23]. It was created in 1997. The concept is making an extension of DNS to check the digital signature [11], thus it provides the basic protection for the privacy.

However, it has some disadvantages. For example, it uses digital signatures therefore the system needs higher performance to process digital signatures. Secondly, the complexity will be highly increased if DNS servers use DNSSEC, then high complexity can cause high possibility to make mistakes [24].

The second soluction is DNSCrypt. Unlike other 3 solutions, it does not follow any RFC, because it was a private standard. The creator is OpenDNS and it was announced in 2011. DNSCrypt does not use digital signatures, it uses cryptographic contruction to encrypt queries [12].

The biggest problem is that it does not follow RFC, which means it is not proposed by IETF. Therefore it is just a private standard, not a public standard, then it is hard to be widely used by DNS providers and application developers. For example, the famous public DNS providers Cloudflare and Google do not support DNSCrypt [23].

The third solution is DOT, it was invented in 2016. The concept is using Transport Layer Security(TLS). TLS is a exist and popular security protocol [13]. Compare with DNSSEC and DNSCrypt, it has a lot of advantages. For example, it does not need a high performance to process encryption. The packet is encrypted, thus it can prevent a man-in-the-middle attack(MITM) [25]. Moreover, DOT follows RFC, therefore it uses a public standard which is proposed by IETF, hence it can be widely used by many DNS providers and developers.

The newest one among those 4 solutions is DOH, it was introduced in 2018, but it is in testing [14]. DOH is similiar to DOT, both DOH and DOT are utilizing TLS to be the tool to encrypt DNS queries. The different is that DOT uses TLS directly, in contrast, there is a protocol between DOH and TLS in the DOH model, which is HTTPS. HTTPS also uses TLS, thus DOH uses TLS indirectly and uses HTTPS directly [1]. The models of DOT and DOH are shown in Table 2.1 and Table 2.2.

DNS (The highest layer)
TLS
TCP
IP (The lowest layer)

Table 2.1: The model of DOT [1]

DNS (The highest layer)
HTTPS (The layer which is different from DOT)
TLS
TCP
IP (The lowest layer)

Table 2.2: The model of DOH [1]

Compare with DOT, DOH uses HTTPS, therefore it is easier to be used than DOT in a browser or other application. DOH server has a domain name of HTTPS, users just input the domain name in their browser then they can start to use DOH.

Even though DOH is still in testing, many public recursive name servers already use it, such as Cloudflare, Google, AdGuard, CleanBrowsing,OpenDNS and Quad9. Meanwhile, Google Chrome, Microsoft Edge, Mozilla Firefox and Opera, those 4 famous browsers also already support DOH.

DOH has resolve many problems that previous solutions may not resolve. It does not require high performance as DNSSEC requires, it follows RFC 8484, thus it is a public standard, unlike DNSCrypt is a private standard. However, base on the situation DOH is still in testing, some issues we may not know. Thus, this paper is going to use DOH server to collect some data to evaluate the functionality, performance and try to find out the problem may have if using a DOH server.

The comparison of different solutions for encrypting DNS queries is shown in Table 2.3.

Solutions	**DNSSEC**	**DNSCrypt**	**DOT**	**DOH**
Introduced	1997	2011	2016	2018
RFC	4033,4034,4035	None	7858,8310	8484

Table 2.3: The solutions for encrypting DNS queries

2.4 The development of Trusted Recursive Resolver

As RFC7626 mentioned [21], the personal data could be revealed in the DNS traffic or DNS servers. About the DNS traffic, it can be encrypted by above 4 solutions in the last section. As for the DNS servers, it need another means to resolve the problem of privacy.

In this background, the concept of Trusted Recursive Resolver(TRR) [9] was designed by Mozilla. TRR is the program for protecting privacy. On the one hand, TRR requires recursive resolvers use DOH to encrypt the content of DNS queries, on the other hand, recursive resolvers have to be supervised. Thus, the personal data both in DNS traffic and DNS servers can be protected from eavesdroppers or taken by others.

There are 3 features in the supervision to recursive resolvers [26].

The first one, the data should be limited. It may be remained in the server only for 24 hours. Moreover, the data can not be sold or shared.

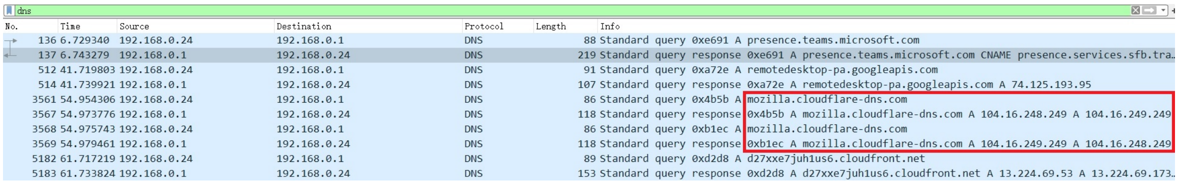
The second one, the data in recursive resolvers need to be transparent. People have the right to understand how the date is stored and used.

The third one, the data in recursive resolvers can not be blocked, filtered, modified, the exception is "required by law".

In case a recursive resolver is able to satisfy those 3 conditions, then it will be "trusted" by users, hence, this recursive resolver is so-called "Trusted Recursive Resolver".

Thus, not every recursive resolver is eligible to join the TRR program. Mozilla selects it and provides a list of trusted recursive resolvers to users. So far, the Internet service provider Comcast and DNS providers Cloudflare and NextDNS have joined the TRR program.

The TRR program has been implemented in Mozilla's browser Firefox. If people enable the TRR function in Firefox instead of using the traditional DNS query, then we can not find any information about browsed websites in packets. The screenshot is shown in Fig. 2.3.



dns

No.	Time	Source	Destination	Protocol	Length	Info
136	6.729340	192.168.0.24	192.168.0.1	DNS	88	Standard query 0xe691 A presence.teams.microsoft.com
137	6.743279	192.168.0.1	192.168.0.24	DNS	219	Standard query response 0xe691 A presence.teams.microsoft.com CNAME presence.services.sfb.tra…
512	41.719803	192.168.0.24	192.168.0.1	DNS	91	Standard query 0xa72e A remotedesktop-pa.googleapis.com
514	41.739921	192.168.0.1	192.168.0.24	DNS	107	Standard query response 0xa72e A remotedesktop-pa.googleapis.com A 74.125.193.95
3561	54.954306	192.168.0.24	192.168.0.1	DNS	86	Standard query 0x4b5b A mozilla.cloudflare-dns.com
3567	54.973776	192.168.0.1	192.168.0.24	DNS	118	Standard query response 0x4b5b A mozilla.cloudflare-dns.com A 104.16.248.249 A 104.16.249.249
3568	54.975743	192.168.0.24	192.168.0.1	DNS	86	Standard query 0xb1ec A mozilla.cloudflare-dns.com
3569	54.979461	192.168.0.1	192.168.0.24	DNS	118	Standard query response 0xb1ec A mozilla.cloudflare-dns.com A 104.16.249.249 A 104.16.248.249
5182	61.717219	192.168.0.24	192.168.0.1	DNS	89	Standard query 0xd2d8 A d27xxe7juh1us6.cloudfront.net
5183	61.733824	192.168.0.1	192.168.0.24	DNS	153	Standard query response 0xd2d8 A d27xxe7juh1us6.cloudfront.net A 13.224.69.53 A 13.224.69.173…

Figure 2.3: *The queried website can not be revealed in packets while using TRR in Firefox*

Enabling the TRR function in Firefox is quite simple, only few steps to enable it. The steps are shown in Fig. 2.4 and Fig. 2.5. Compare with DOT, DOT can run on Android and IOS, but it needs some tools to run on Linux and Windows, thus it is more inconvenient [13].

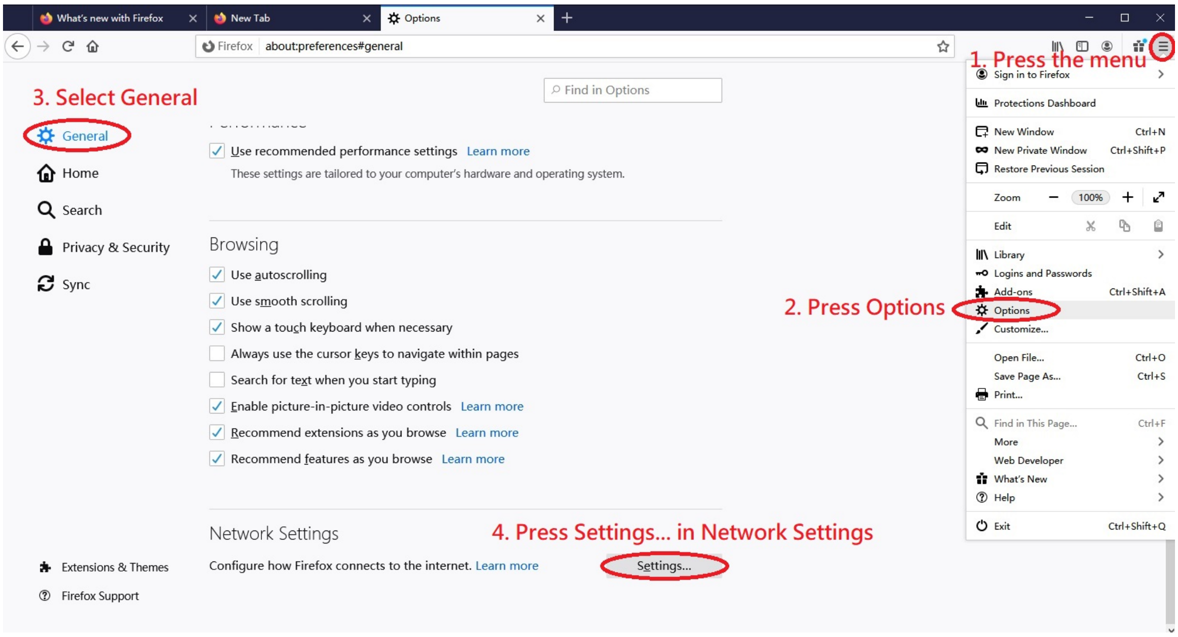


Figure 2.4: *The steps to enable TRR in Firefox - Part 1*

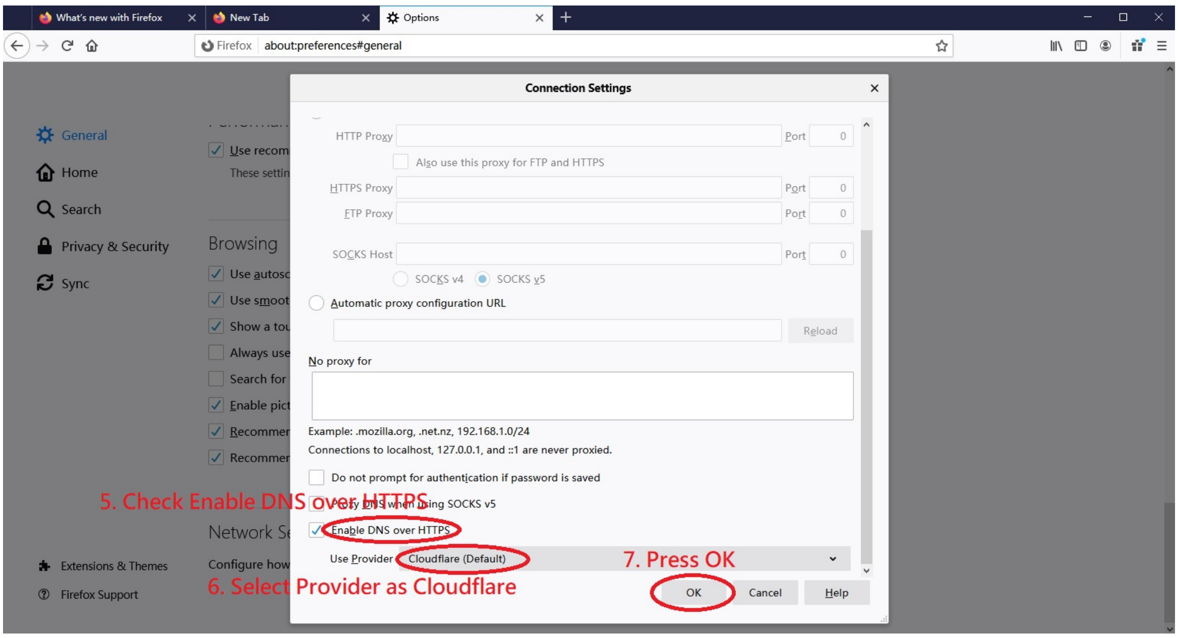


Figure 2.5: *The steps to enable TRR in Firefox - Part 2*

Furthermore, Mozilla is also devoted in pushing TRR. In February 2020, Mozilla set TRR as the default setting of Firefox in USA, which means Firefox users used the DOH service if they do not do any change [27].

However, this policy caused a severe problem, after the TRR function was set as default setting in Firefox, the cooperative DOH DNS providers Cloudflare and NextDNS suffered a very high workload and they were hard to work on it. It seems like a kind of DDOS attacking to those DOH DNS providers, therefore Mozilla had to change the policy. In the later version Firefox 77.0.1 in June 2020, TRR has been removed the default setting, now users have to set the TRR function by themselves manually if they wish to enjoy DOH service [28].

Hence, The time TRR been default setting was only lasting about 4 months(from February 2020 to June 2020). If Mozilla wants to make TRR being the default again, then the performance of DOH DNS servers must be capable to process the queries of national scale.

As for Google Chrome, the main opponent to Mozilla Firefox, Chrome also supports DOH, and Google also intended to adjust DOH service to be the default setting for DNS queries as well [29].

The difference is that, the policy Google adopted is not TRR, because TRR is the unique program to Mozilla and its browser Firefox. The policy Google adopted is automatic upgrade. Which means, if the criteria that users have is met the requirement for using DOH service, then Chrome will change the setting to be DOH and users do not need to do anything. Google has already implemented this function in Chrome version 83 in May 2020 [30].

Another difference is that Chrome can not customize a DOH server in the browser , the options are only default, enabled and disabled in the DOH setting. Users have to set a DNS server which supports DOH in their operating systems. Contrarily, Firefox is more flexible, users are able to set a DOH DNS server which is different from the DNS server in their operating system. Therefore, it is easy to test or use a private DOH DNS server [31].

The interface of DOH setting is shown in Fig. 2.6.

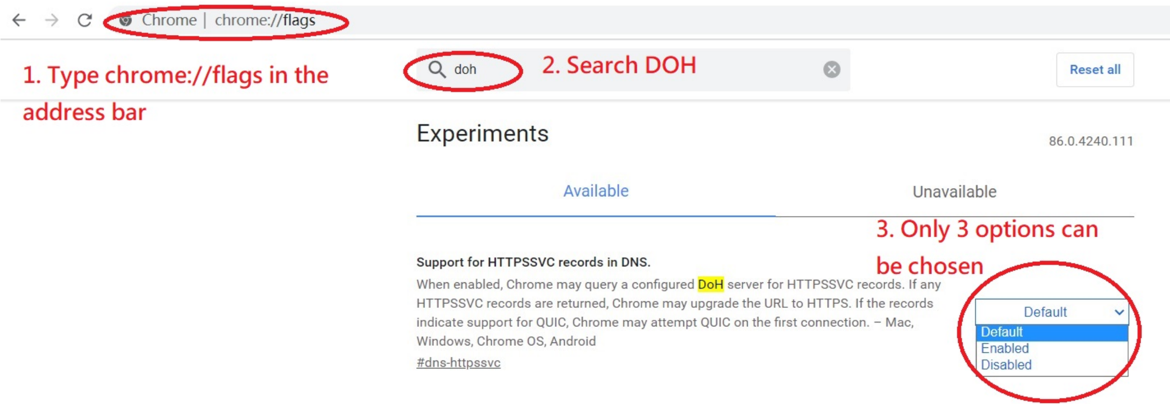


Figure 2.6: *DOH setting on Google Chrome*

Not only Firefox and Chrome support DOH, but also other browsers, such as Opera [32], Microsoft Edge and Vivaldi [33]. Every companies or browser has their own plans to support DOH, those plans may be different and have their features, advantages and disadvantages.

However, this study just focuses on the TRR program which is executed by Mozilla Firefox. Thus, in a later chapter, this study is going to utilize the TRR function in Firefox to test a private DOH DNS server, which was built by the researcher, to figure out how the performance will be.

Chapter 3

The DNS traffic in Ireland

3.1 The introduction of root servers

The DNS traffic is an important consideration for building a DNS server. Irish TRR servers have to be capable to deal with the DNS traffic of national scale traffic in Ireland.

Before understanding the DNS traffic in Ireland, it is necessary to understand the root servers first.

Root servers are the highest level DNS servers [5], there are 1097 instances in the root server system on 31 August 2020. They are divided into 13 root server zones, each zone has a representative letter, which are A, B, C, D, E, F, G, H, I, J, K, L and M [34]. Those root server zones are managed by 12 organizations [2], which are Verisign(It manages 2 root server zones), USC-ISI, Cogent Communications, University of Maryland, NASA Ames Research Center, Internet Systems Consortium, Defense Information Systems Agency, U.S. Army Research Lab, Netnod, RIPE NCC, ICANN and WIDE Project. Therefore, those 12 organizations have the information about DNS traffic.

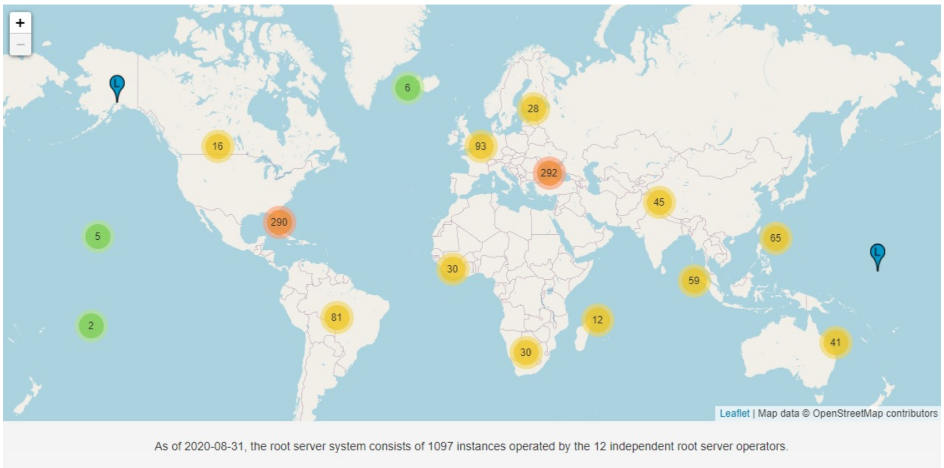


Figure 3.1: *The root servers in the world [2]*

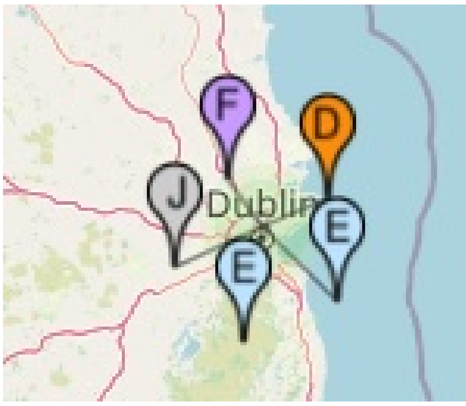


Figure 3.2: *The root servers in Dublin [2]*

Figure 3.3: *The root servers in Cork [2]*

In the map from Root-servers.org, there are 8 root servers in Ireland, 5 servers are in Dublin and 3 servers are in Cork. As for organizations, 3 servers belong to E-root(E zone, it is managed by NASA Ames Research Center), 2 servers belong to F-root(F zone, it is managed by Internet Systems Consortium). K-root(RIPE NCC), D-root(University of Maryland), J-root(Verisign) have 1 server respectively [2].

Zone	Operator	Number in Ireland
A	Verisign	0
B	USC-ISI	0
C	Cogent Communications	0
D	University of Maryland	1
E	NASA Ames Research Center	3
F	Internet Systems Consortium	2
G	Defense Information Systems Agency	0
H	U.S. Army Research Lab	0
I	Netnod	0
J	Verisign	1
K	RIPE NCC	1
L	ICANN	0
M	WIDE Project	0

Table 3.1: The list of root server zones [2]

The problem is the information those organizations provided on Internet is limited. There is no statistic data of DNS queries in Ireland on their website.

It is very hard to collect the records about all DNS queries, because there are numerous recursive DNS servers and they are managed by many organizations [35], hence, the operators of root servers do not have their data.

3.2 The first method to estimate Irish DNS traffic

However, it is not necessary to know the total number of DNS queries in Ireland, because the target in this study is to understand what the performance should a TRR server possess in Ireland. The number of DNS queries a root server received may help us to evaluate the required performance for a TTR server in Ireland.

In this paper, the researcher designs some methods to estimate the traffic a DNS server would have in Ireland by using the traffic in root servers.

Method 1 is using the data on Akamai.com to estimate the DNS traffic [3].

In website Akamai.com, it collects the DNS traffic from 9 root server zones(B, C, D, E, F, I, K, L, M), but the DNS traffic is worldwide, it does not provide the data in national scale or city scale on the website.

Even though there is no national scale data on Akamai.com, but the worldwide data can be used to estimate the Irish DNS traffic.

In a report from Central Statistics Office of Ireland, it showed that there were 89% of Irish households have the internet at home in 2018 [36]. From the growth of households with the internet, the percentage is probably 90% in 2020. There were about 4.57 billion internet users in the world in July 2020 [37]. The population in Ireland was around 4.944 million in August 2020 [38]. Hence, the Irish Internet users may be about 4.113 million, it was approximately 0.09% of internet users in the whole world.

According to the data from Akamai.com [3], the overall DNS traffic in the world was about 7 Trillion transactions (Requests and responses) in June 2020. Then, 0.09% of DNS traffic in the world could be Irish DNS traffic, which is around 6.3 billion DNS transactions for one month in Ireland. On average, it could be 210 DNS million transactions in a day in Ireland.

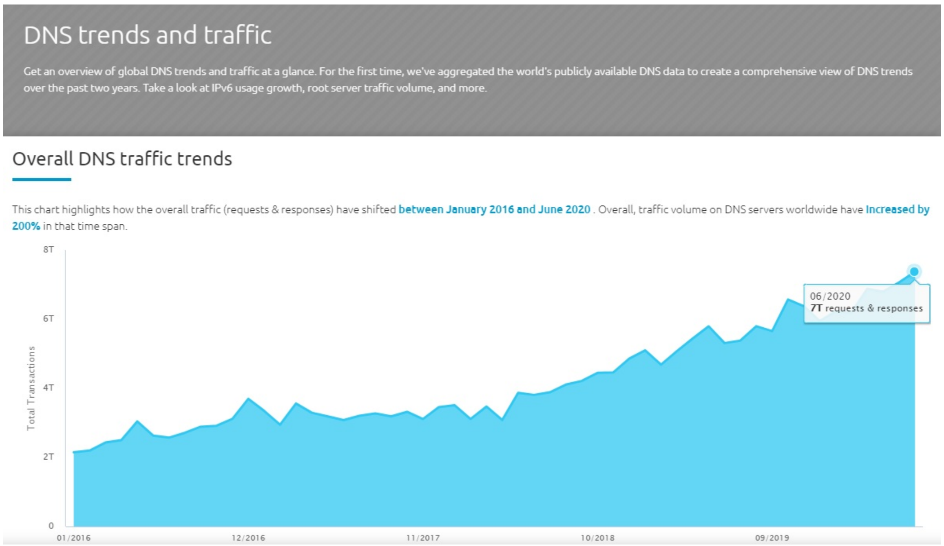


Figure 3.4: *The trend of DNS traffic in the world [3]*

Month	IPv4	IPv6	Total
06/2020	6T	1T	7T
01/2020	5T	969B	6T
07/2019	4T	919B	5T
01/2019	4T	848B	5T
07/2018	3T	564B	4T
01/2018	3T	426B	4T
07/2017	3T	371B	3T
01/2017	3T	363B	3T
07/2016	2T	248B	3T
01/2016	2T	171B	2T

Table 3.2: Overall DNS traffic trends(Unit:Transactions) [3]

However, internet traffic is changeable in different hours, it is necessary to understand when are the rush hours. For example, the internet rush hours are usually between 7 pm and 11 pm in UK [39]. In Sao Paulo, the internet rush hours are between 8 pm and 11 pm [40]. In USA, it is 8 pm to 10 pm [41]. In Berlin, the rush hours are

8 pm to 11 pm [42]. In Amsterdam, it is from 8 pm to 11 pm as well [4].

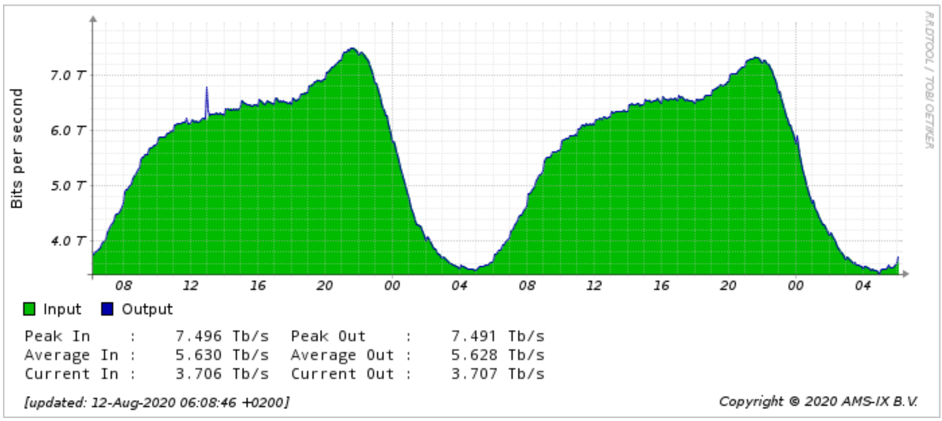


Figure 3.5: *The internet traffic in a day (Amsterdam) [4]*

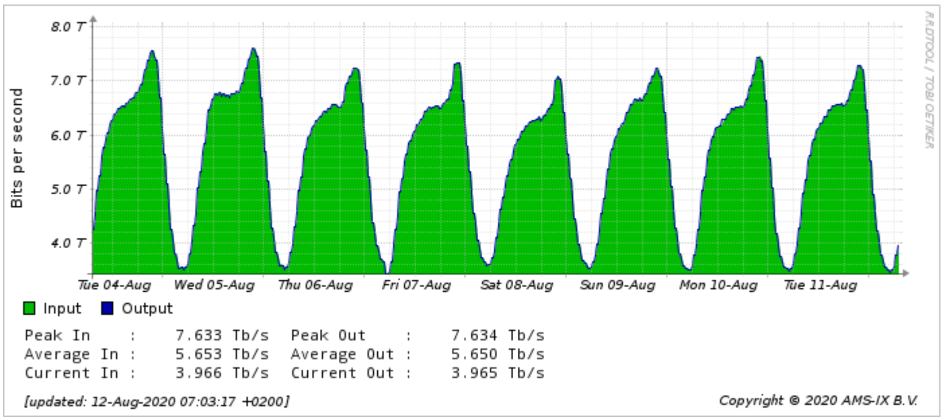


Figure 3.6: *The internet traffic in a week (Amsterdam) [4]*

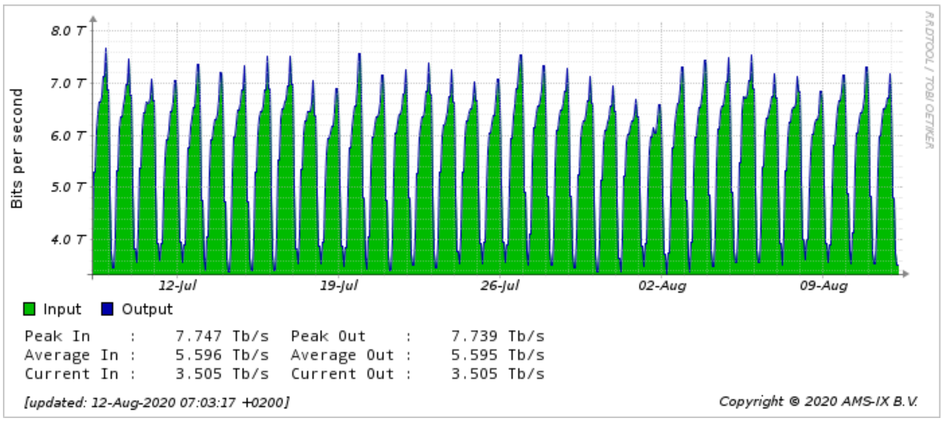


Figure 3.7: *The internet traffic in a month (Amsterdam) [4]*

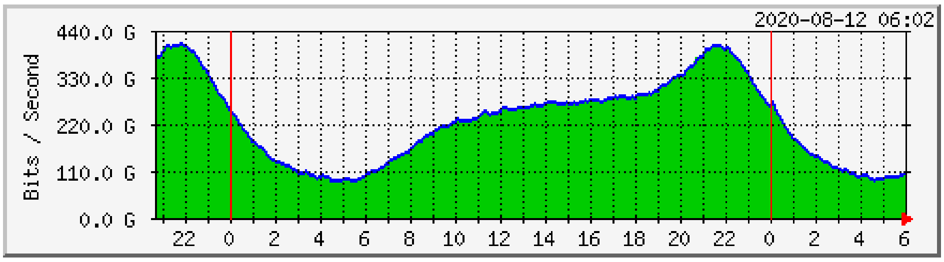


Figure 3.8: *The internet traffic in a day (Berlin) [4]*

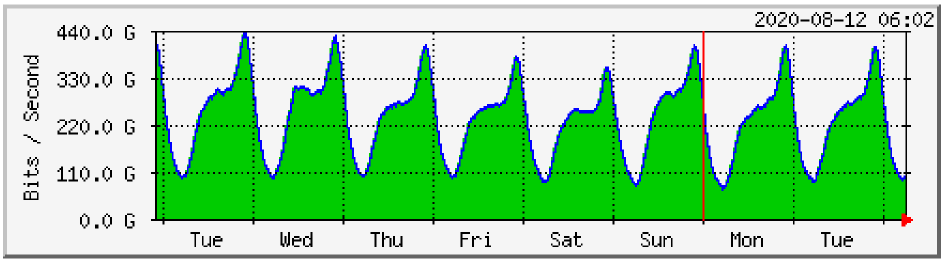


Figure 3.9: *The internet traffic in a week (Berlin) [4]*

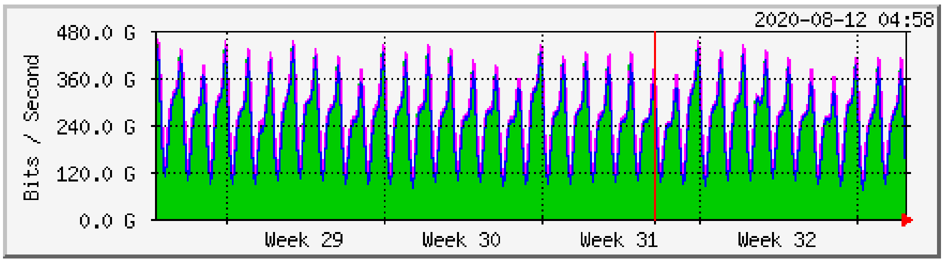


Figure 3.10: *The internet traffic in a month (Berlin) [4]*

All the reports in different countries or cities revealed that internet rush hours are from 8 pm to 11 pm, the distributions are pretty similar. Therefore, Irish internet rush hours could be assumed as from 8 pm to 11 pm as well.

As for the comparison in different days in a week, from Monday to Sunday, the change is not obvious. About the days in a month, from the begin to the end of a month, there is no huge difference as well.

Taking the data in Amsterdam to estimate the percentage of usage in each hour, the result is shown in TABLE 3.3.

Hour(24)	Trillion bit/s	Percentage
0	5.8	4.3%
1	4.8	3.56%
2	4	2.96%
3	3.7	2.74%
4	3.6	2.67%
5	3.5	2.59%
6	3.6	2.67%
7	4	2.96%
8	4.8	3.56%
9	5.4	4%
10	5.8	4.3%
11	6	4.44%
12	6.2	4.59%
13	6.4	4.74%
14	6.4	4.74%
15	6.6	4.89%
16	6.6	4.89%
17	6.6	4.89%
18	6.5	4.81%
19	6.7	4.96%
20	6.9	5.11%
21	7.2	5.33%
22	7.2	5.33%
23	6.7	4.96%
Total	135	100%

Table 3.3: Internet traffic and its percentage in each hour in a day in Amsterdam [4]

After that, using the percentage to multiply the estimated number of daily DNS transactions in Ireland, which is 210 million, then the result is sown in TABLE 3.4. The 2 busiest hours are 9 PM and 10 PM, the number of DNS transactions could be

11.2 million in an hour.

Hour(24)	**Percentage**	**Million transactions**
0	4.3%	9.02
1	3.56%	7.47
2	2.96%	6.22
3	2.74%	5.76
4	2.67%	5.6
5	2.59%	5.44
6	2.67%	5.6
7	2.96%	6.22
8	3.56%	7.47
9	4%	8.4
10	4.3%	9.02
11	4.44%	9.33
12	4.59%	9.64
13	4.74%	9.96
14	4.74%	9.96
15	4.89%	10.27
16	4.89%	10.27
17	4.89%	10.27
18	4.81%	10.11
19	4.96%	10.42
20	5.11%	10.73
21	5.33%	11.2
22	5.33%	11.2
23	4.96%	10.42
Total	100%	210

Table 3.4: Using the daily distribution of Internet traffic of Amsterdam to estimate the DNS traffic in Ireland [4]

However, the data from Akamai.com does not include A, G, H and J root server zones. Thus, the number of transactions in rush hours should be higher than 11.2 million.

The numbers of transactions in every root server zone are very different, therefore it is hard to estimate the numbers in A, G, H and J root server zones. If assume that the average number of A, G, H and J is close to the average number of B, C, D, E, F, I, K, L, M, then the estimated number of transactions in all root servers in rush hours could be $11.2/9\times13=16.18$ million.

If convert it into a second, the number of DNS transactions could be 4,494 per second during the rush hour (16.18(million per hour)/60(minutes)/60(seconds)).

3.3 The second method to estimate Irish DNS traffic

Method 2 is using the data from ICANN to estimate DNS traffic.

ICANN (Internet Corporation for Assigned Names and Numbers) is the one of 12 organizations which are responsible for managing DNS root servers, the servers it manages are L-root servers. Unlike other 11 organizations, ICANN provides a website to display real-time DNS traffic, that is Stats.dns.icann.org [6].

The problem is ICANN does not have root servers in Ireland. Moreover, those data is only from ICANN, it does not include the data from other root server zones.

Thus, here chose a city which has a similar population to Ireland to estimate the DNS traffic. Melbourne should be a ideal sample. The population in Melbourne is about 5 million in 2019 [43], which is close to the population in Ireland (4.9 million). Moreover, Melbourne is isolated, there is no big city near Melbourne,therefore the network connection may be similar to a country.

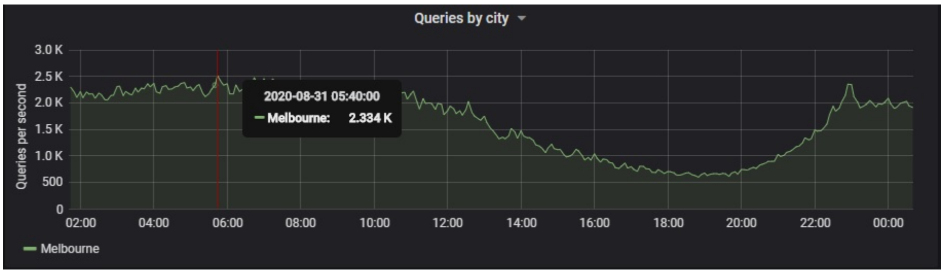


Figure 3.11: *The number of queries per second in a root server in Melbourne [6]*

From the data in Fig. 3.11, the highest value in a day is 2,334 per second, which was occurred at 05:40 UTC(19:40 in Melbourne).

Figure 3.12: *The root servers in Melbourne [2]*

However, the data is only from a root server which is managed by ICANN, there are four root servers in Melbourne, the DNS traffic in other 3 root servers are not provided. Assume the average traffic in other 3 root servers is close to the root server managed by ICANN, the whole DNS traffic in Melbourne could be 9,336 per second(2,334X4). Next, adjust the traffic to accord the population of Ireland, the DNS traffic could be 9,149 per second during the rush hour(9,336/5X4.9).

The comparison between method 1 and method 2 is shown in TABLE 3.5.

	Method 1	**Method 2**
Source	Akamai.com	ICANN
Queries per second in rush hours	4,494	9,149
Drawback 1	No Irish data	No Irish data
Drawback 2	Only monthly data	The data is only from L-root

Table 3.5: The comparison between 2 methods for the estimation of the DNS traffic in Ireland

Chapter 4

The required software

4.1 The overview of required software

There are 2 kinds of DNS servers, authoritative DNS servers and recursive resolvers. The DNS software we talk about here is the software to build a recursive resolver. There are many choices to implement a DNS server, such as BIND, Unbound, DNSMASQ, PowerDNS, Microsoft DNS and Cisco Network Registrar [44] [45].

In this study, Unbound, BIND and PowerDNS are recommended, because there are many discussions and tutorials about those three software on Internet [46] [47] [48]. Thus, people may be easy to use them to build private or public recursive resolvers.

Unbound is the free open-source software which focuses on building a recursive DNS server, it does not support the authoritative DNS server. The developer is NL-net Labs, the developer also designed another DNS software, which is NSD(Name Server Daemon), in contrast, NSD is only for building authoritative DNS servers [49]. Unbound supports some security functions, such as Domain Name System Security Extensions(DNSSEC) and DNS over TLS(DoT). Moreover, the operating systems for running Unbound can be Linux, FreeBSD and Windows [50].

About BIND, its alias is Named. Unlike Unbound, it supports both recursive and authoritative DNS servers. It is developed by Internet Systems Consortium(ISC). ISC is also the organization which is responsible for managing F root server zone. The stable version is BIND 9. It can run on Windows, Mac-OS and Linux [51].

PowerDNS, it supports both authoritative DNS server and recursive DNS server, moreover, it provides a Graphic UI for management and uses relational databases to store data. The developer is PowerDNS Community and operating systems are Linux and FreeBSD [52].

Apart from DNS software, building a DNS server for TRR also needs other some software, including operating systems, DOH tools and HTTP servers [53] [46].

The DOH DNS server need DOH tools to receive DoH queries and test. doh-proxy is designed for this purpose, the developer is Facebook. It can be installed on Linux but it requires Python 3.5 [54]. DOH-proxy inlcudes 4 tools, which are doh-proxy, doh-httpproxy, doh-stub and doh-client. doh-httpproxy and doh-client were used in this study, because doh-httpproxy provides the DOH service, and doh-client is the testing tool to connect a DOH DNS server to check the installation is successful or not.

After that, NGINX can provide the HTTP service. NGINX is a HTTP server with high performance, it can also provide different kinds of services. The operator can set the method for listening queries from users [55].

In choosing a operation system, there are many operating systems could be used to install DNS servers, such as Windows server, FreeBSD, Linux.

There are many members in the Linux family, including Fedora,Red Hat Linux, CentOS, Ubuntu, Debian [56]. CentOS was chosen for testing here, because it is free, and the structure is the offshoot of Red Hat enterprise, hence it is very stable.

The required software is shown in TABLE 4.1.

Category	Software	Note
DNS	BIND	
DNS	Unbound	Free and open-source software
DNS	PowerDNS	
DOH Tool	doh-proxy	
HTTP Server	NGINX	Free and open-source software
Operating System	Linux	Windows can be used as well

Table 4.1: The required software for building a DNS server for TRR

4.2 The comparison and installation among DNS software applications

In order to understand the situation of the setting and usability among BIND, Unbound and PowerDNS, the researcher tried to install those 3 DNS software on a server and test them.

Due to the effect of COVID-19, students were not allowed to go inside the laboratory, hence this study has to be finished at home, therefore the equipment was limited, only 4 personal devices were available to the researcher, which were a personal desktop computer, 2 android phones and a IPad. The operating system on the personal desktop computer is Windows 10. Thus, there was no spare computer to install Linux, the researcher had to use a virtual machine to install Linux on the personal computer. The software for running virtual machine was VMware Workstation Player.

After installing the operation system, then installed BIND, Unbound and PowerDNS, and set the configuration files of those DNS servers, to make sure those DNS servers were in the same network zone with other testing devices.

Next, used Internet Information Services(IIS) to create a simple website on the personal computer, and gave fixed local IP addresses to all devices and the virtual machine, then both the DNS server and website had the IP addresses. In the DNS server(BIND, Unbound and PowerDNS), set a domain name to the simple website to match its IP address. The simple website was the testing website.

Finally, used the testing devices(Android phone and IPad) to type the domain name of the testing website on browsers(Google Chrome and Safari). If the testing website could be displayed on testing devices, then the DNS server functioned well.

Above steps were the testing method for the study. The testing environment is shown in TABLE 4.2.

Platform	VMware Workstation 15.5.6 Player
Operating system(Server)	CentOS 8.2.2004
Internet connection	Bridge mode
Testing devices	Desktop, IPad, Android phone
Operating system(Client)	Windows, IOS, Android
Testing method	Local website browsing

Table 4.2: The installation environment

Next, the following discussion is about configuration. The configuration of BIND is using C language to be the format of the configuration file. Therefore, in case the maintenance personnel does not have programming background, then it needs time to understand the syntax of C language.

About Unbound, the format of configuration file is very simple, it does not belong to any kind of computer languages. The setting is just listed line by line.

PowerDNS adopts relational database, thus the configuration of PowerDNS has 2 parts, the first part is the normal configuration file, it decides the setting for the database. The second part is in the database, the contain is including domains and records.

The installation of PowerDNS is more complex than Unbound and BIND, because it uses the rational database. However, it is a double-edged sword, it is not friendly for normal users during the installation, but after installation, PowerDNS provides the website interface to display the information of the DNS server, and the website is also the interface for the setting, hence the maintenance is easier than BIND and Unbound.

The comparison among BIND, Unbound, PowerDNS is shown in Table 4.3.

	BIND	**Unbound**	**PowerDNS**
Version	9.11.13	1.7.3	4.3.1
Configuration	C language	Line by line	RDBMS
Log style	Log file	Log file	MySQL
Installation difficulty	Easy	Easy	Difficult
Maintenance difficulty	Normal	Normal	Easy

Table 4.3: The comparison among BIND, Unbound, PowerDNS

4.3 Previous studies about the performance of DNS software

According to a report which was written by Hamza Boulakhrif in University of Amsterdam, the times for processing queries in BIND, Unbound and PowerDNS were similar. The biggest difference was that when the time of processing exceeded 16 seconds, then PowerDNS did not response. If the time exceeded 17 seconds, BIND did not response, only Unbound can wait until the finish of processing then sent results to users [7].

The processing time in those 3 DNS software in this report is shown in Fig. 4.1.

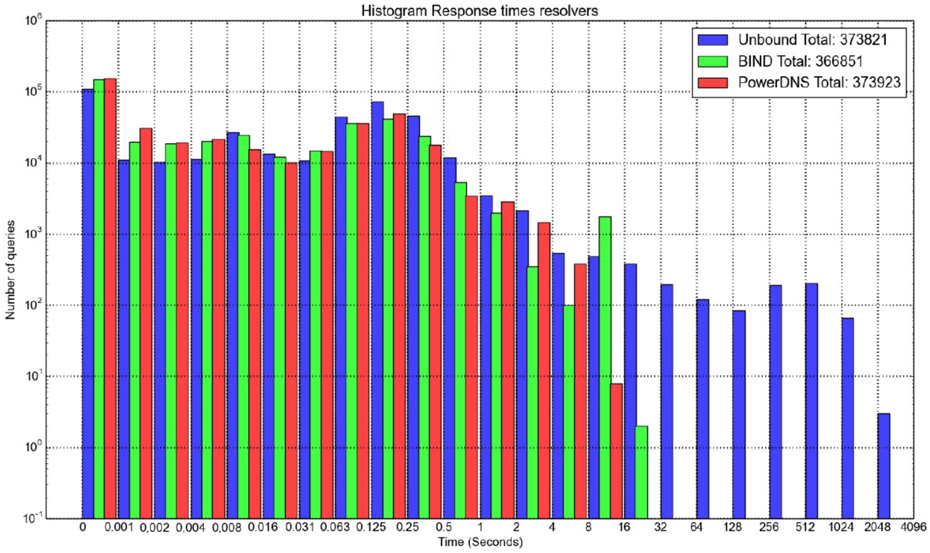


Figure 4.1: *The processing time in BIND, Unbound and PowerDNS [7]*

In Fig. 4.1, according to the assumption of Hamza Boulakhrif, if processing time is under 1 milliseconds, then it is processed by the cache, because the caches in DNS servers contain the matched IP addresses that users are looking for, thus DNS servers can response users in a very short time and do not need to ask authoritative DNS servers.

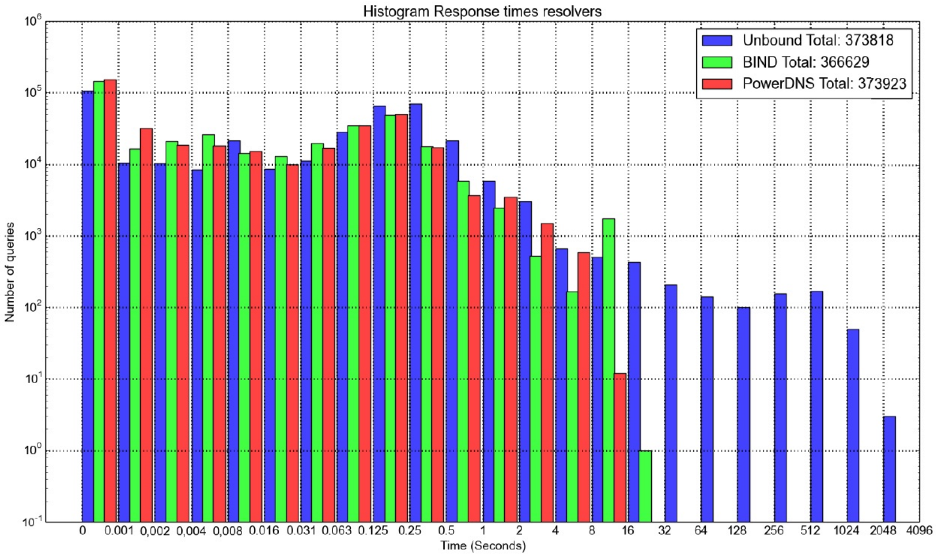


Figure 4.2: *The processing time in BIND, Unbound and PowerDNS with DNSSEC [7]*

In the other experiment, BIND, Unbound and PowerDNS ran with DNSSEC, there was no obvious change if compare it with these DNS servers without DNSSEC. The result of using DNSSEC is shown in Fig. 4.2.

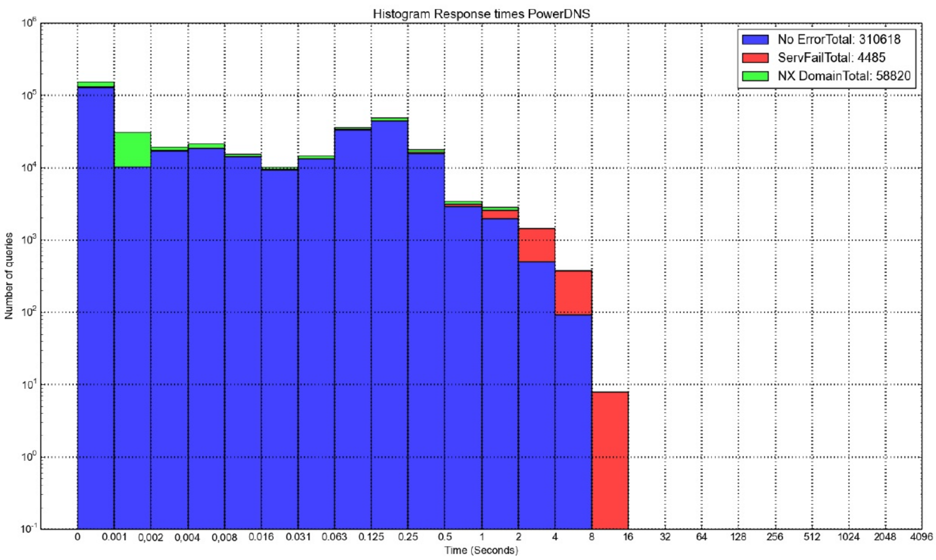


Figure 4.3: *The processing time in PowerDNS [7]*

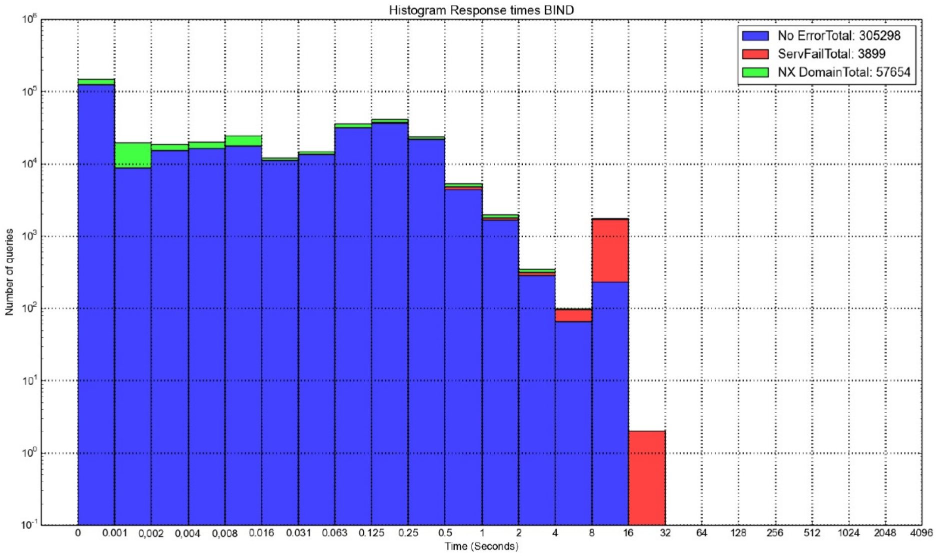


Figure 4.4: *The processing time in BIND [7]*

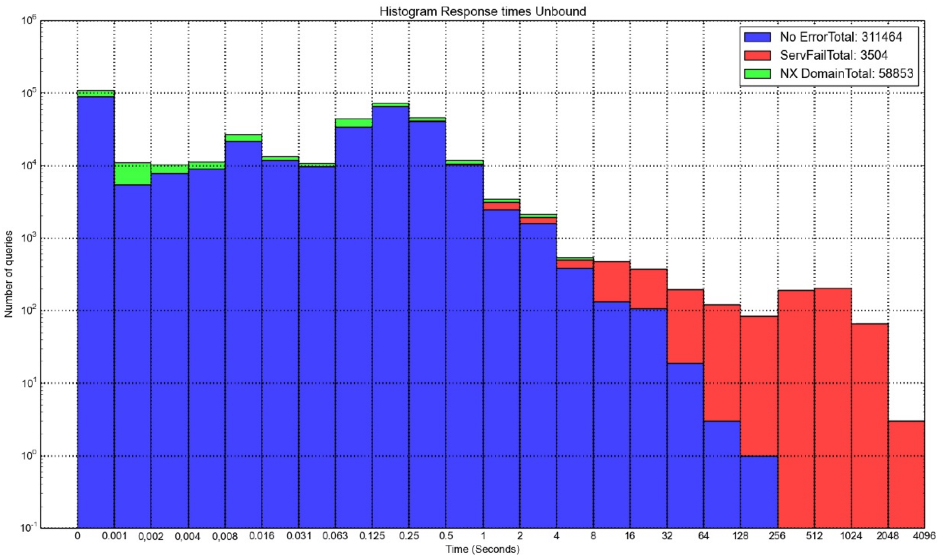


Figure 4.5: *The processing time in Unbound [7]*

Fig. 4.4, Fig. 4.5 and Fig. 4.3 show the details of the time of processing queries in BIND, Unbound and PowerDNS. The working types in BIND and PowerDNS are similar, both of them have the time limit, thus in case the processing time reaches the time limit, then the query will be failed. In contrast, Unbound has a different working type, it allows the system to have a long time to wait for the answer after the process. Hamza Boulakhrif called those different working types as "Failed response over a late response" and "An answer is better than no answer". PowerDNS adopts "Failed response over a late response" and Unbound adopts "An answer is better than no answer", as for BIND, it is between PowerDNS and Unbound [7].

In conclusion, in choosing DNS resolver software, because BIND, Unbound, PowerDNS have similar performances, therefore the point of the decision is allowing a long time to wait for the answer or not. If yes, the operator should choose Unbound. Otherwise, he should choose BIND or PowerDNS.

However, the report was written in 2015, which was 5 years ago, the performance may be different now.

Chapter 5

The experiment

5.1 The implementation of the experiment

In order to understand the performance of a DOH server, the researcher designed an experiment. First of all, a DOH server was built in the local network. After that, used Python to design a testing program. The function of the testing program was sending massive DNS queries to the DOH server in a very short time. Then, the testing program recorded results and latencies of the responses from the DOH server. The testing environment is described in Table 5.1.

The testing program utilized multi-thread to send queries, thus queries can be sent in a very short time. 50 to 500 queries can be sent between 0.2 to 0.4 seconds. The outputs of this testing program are CSV files, those CSV files contain the records and statistic of the tests in this experiment.

	Description
Server platform	VMware Workstation 15.5.6 Player
Server operating system	Linux CentOS 8.2.2004
Client operating system	Windows 10
Server CPU	Intel Core i7-7700(3.60GHz, 4 cores
Server memory	8 GB
Server hard-disk	20 GB
DNS resolver	BIND 9.11.13
DNS tools	DOH-proxy
HTTPS server	Nginx 1.14.1
Testing language	Python 3.9
DNS library	DNSPython, dnslib(paulc)
Other library	ssl, csv, json, base64, urllib, threading

Table 5.1: The testing environment of the experiment

Meanwhile, the testing program can set some parameters, therefore the researcher was able to gain the data from different parameters, those parameters were independent variables or controlled variables [57] in this experiment. The variables are shown in Table 5.2. The independent variables were the changed parameters and the controlled variables were the unchanged ones.

Independent variables	**Options**
Cache	Flushed cache, Non-flushed cache
Query name	Top 50 websites in Ireland, top 500 websites in the world
Record type	A, AAAA, CNAME, MX
DNS type	DOH(Wireformat), DOH(JSON), Traditional DNS
DNS provider	Local(Built by the researcher), Cloudflare, Google
Query interval	No interval, 0.1 Sec., 0.01 Sec. 0.001 Sec.
Query time	Peak time, Off-peak time

Table 5.2: The independent/controlled variables(parameters) in the experiment

The outputs of the testing program were dependent variables, which were the results of the experiment. The researcher used those results from different parameters to analyze the performance of a DOH DNS server, those data can be used to conclude how many DOH DNS servers should have in Ireland. The dependent variables(output) are shown in Table 5.3.

There are 4 results of responses, which are success, NXDomain, no answer, and others. Success is that the query is successful to get the IP address from the response from the DNS server. NXDomain stands for Non-existent Internet domain name [58], which means the DNS server can not find the matched IP address for the queried domain name. The third result is no answer, if the DNS server respond any error message or no message, then the result is classified as no answer. The last one is others, if the query can not reach the DNS server, or the DNS server does not return the response, or any other situations happen, then those results are others.

The result "Others" was unusual, in the most tests, there was no others, only success, NXDomain, and no answer happened. The exception was testing the traditional DNS server, because the case timeout could happen, then it was categorized as others. Thus, in following sections, the graphs do not display the result "other" if the number of others was 0.

Dependent variables	**Outputs**
The results of responses	Success, NXDomain, No Answer, Others
The number of responses	A number for each result, from 0 to 500
The fastest response of a result	Seconds and the domain name
The latest response of a result	Seconds and the domain name
The average response of a result	Seconds
Responses in 0.1 Sec.	A number for each result, from 0 to 500
Responses between 0.1 and 1 Sec.	A number for each result, from 0 to 500
Responses between 1 and 5 Sec.	A number for each result, from 0 to 500
Responses more than 5 Sec.	A number for each result, from 0 to 500
Queries start time	Time
Queries end time	Time
Queries sending duration	Seconds

Table 5.3: The dependent variables(outputs) in the experiment

5.2 The test for the cache

The experiment was separated into different small tests to test the effect from each variable.

The first test was cache. DNS servers use cache to save the IPs of used queries. If the user sends an used query again, the DNS server does not need to ask the authoritative DNS server again, just response the IP in the cache immediately, then it could save a lot of time. The controlled variables are shown in Table 5.4.

Controlled variables	**Parameters**
Query name	Top 50 websites in Ireland
Record type	A
DNS type	DOH(Wireformat)
DNS provider	Local(Built by the researcher)
Query interval	No interval
Query time	From 23:18:59 to 23:19:54 25/10/2020

Table 5.4: The controlled variable for testing the cache

The testing steps were that, firstly, typed "rndc flush" in Linux to flush the cache of BIND, then there was no IP record in the DNS server. Next, ran the testing program to gain the data with the flushed cache. After that, the IP records existed in the cache of DNS, then ran the testing program again to get the data without flushing cache (Non-flushed cache). The distribution of results is displayed in Fig. 5.1 and Fig. 5.2.

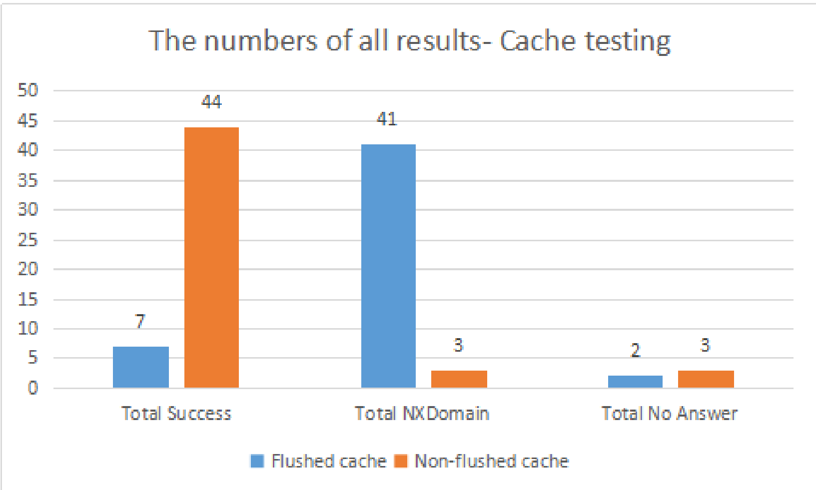


Figure 5.1: *The numbers of each result with flushed cache and non-flushed cache.*

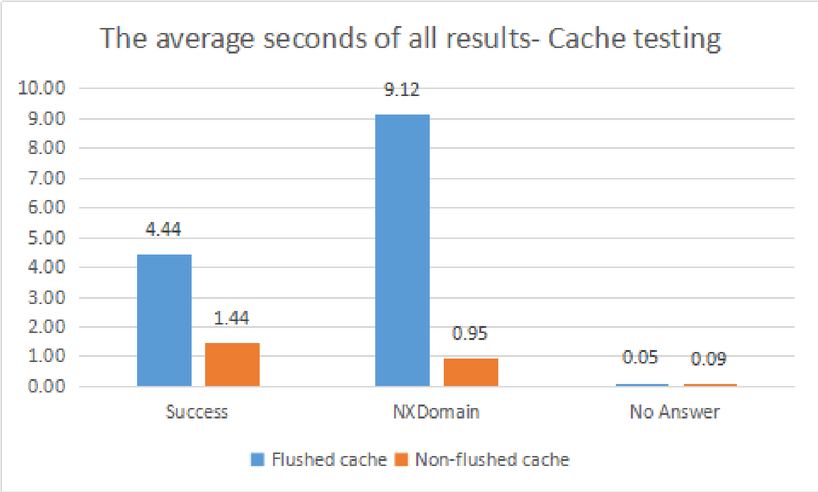


Figure 5.2: *The average seconds of the responses with flushed cache and non-flushed cache.*

The results showed that it was a huge difference, the almost responses with non-flushed cache were successful, contrarily, the almost responses with the flushed cache were failed. As for the average time of responses, the flushed cache also needed several times the time more than the non-flush cache to respond the client.

5.3 The test for 500 worldwide domain names

..

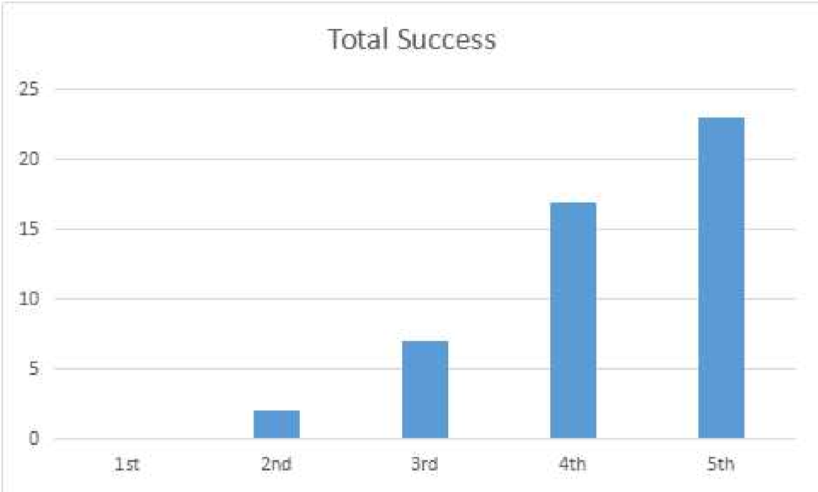


Figure 5.3: *The results of testing top 500 worldwide domains.*

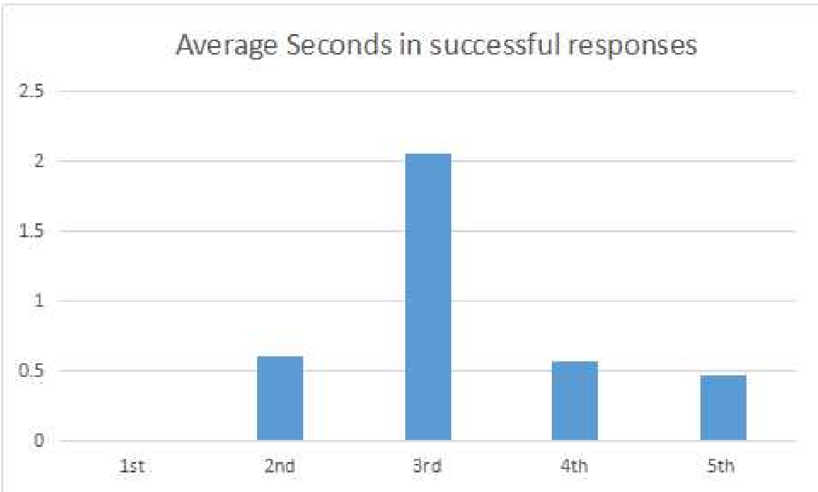


Figure 5.4: *The average seconds of testing top 500 worldwide domains.*

	A	AR	AS
1	Filename	Latest Success Duration	Latest Success Domain
2	1st	0	
3	2nd	0.940847	cloudflare.com
4	3rd	10.292023	istockphoto.com
5	4th	1.082419	sites.google.com
6	5th	1.114272	marketingplatform.google.com

Figure 5.5: *The latest responses of testing top 500 worldwide domains.*

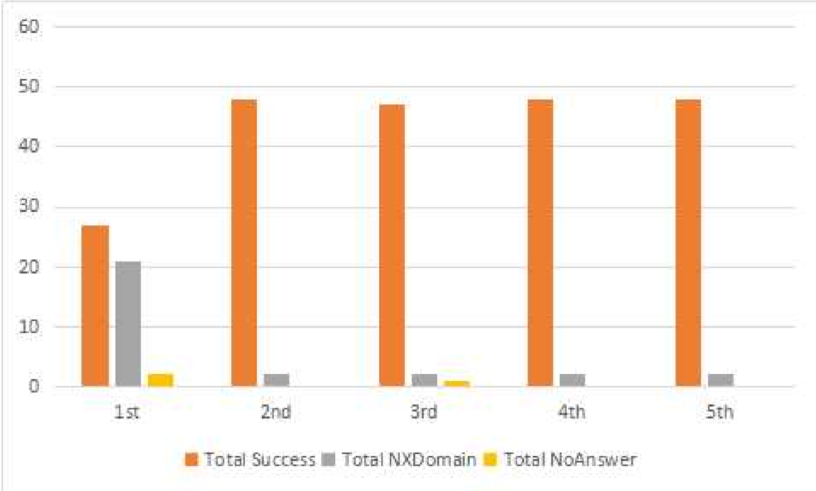


Figure 5.6: *The results of testing top 50 worldwide domains.*

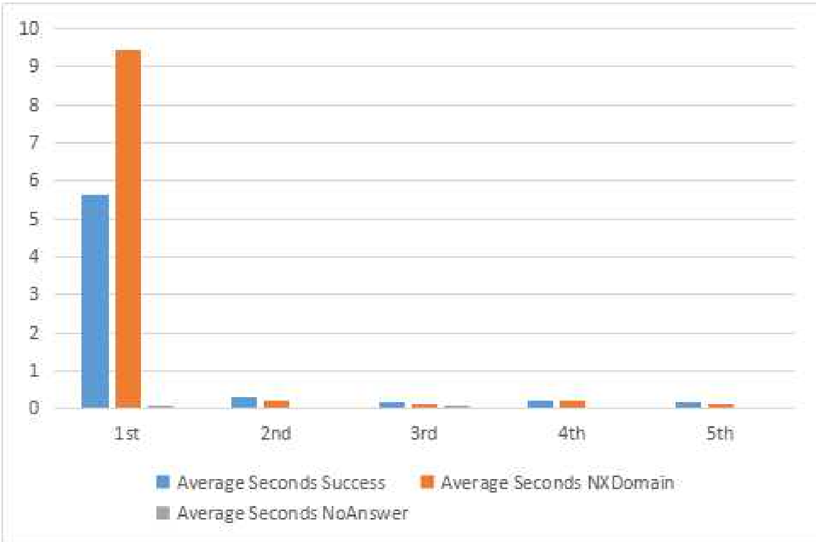


Figure 5.7: *The average seconds of testing top 50 worldwide domains.*

	A	AR	AS	AT	AU	AV
1	Filename	Latest Success Duration	Latest Success Domain	Latest NXDomain Duration	Latest NXDomain Domain	Latest NoAnswer Duration
20	1st	9.882981	youtu.be	10.081403	www.blogger.com	0.043329
21	2nd	1.704705	line.me	0.295512	bp.blogspot.com	0
22	3rd	0.229482	dropbox.com	0.183845	googleusercontent.com	0.041005
23	4th	0.256598	vimeo.com	0.256597	bp.blogspot.com	0
24	5th	0.309713	adobe.com	0.197738	bp.blogspot.com	0

Figure 5.8: *The latest responses of testing top 50 worldwide domains.*

5.4 The test for DNS types

...

5.5 The test for DNS providers

...

5.6 The test for query intervals

...

5.7 The test for query time

...

Chapter 6

Overall design

6.1 The concern of DDOS attacks

DDOS is the important issue for building DNS server [59].

There are 2 sorts of DNS queries, recursive and iterative. At the beginning, users send queries to recursive servers, when recursive DNS servers receive requests, if they do not have the matched IP addresses, then recursive DNS servers can help users to ask authoritative DNS servers for getting IP addresses, then return results to users, that is the recursive query.

As for the iterative query, when authoritative DNS servers receive the queries from recursive DNS servers, if they do not have the matched IP addresses, they will give recursive servers the IP addresses of other authoritative DNS servers for querying, then recursive servers will ask other authoritative DNS servers, this type of querying is the iterative query [60].

However, the recursive queries may cause DDOS attacks. The content of packet could be faked, the IP address of a sender can be changed to be the IP address of the victim. In case thousands of computers send recursive queries to DNS servers, and all IPs of sources are changed to be the IP of a victim, then those DNS servers will send thousands of responses to that victim. After that, the traffic in the victim would be too high then cause some problems [8]. This type of DDOS attack is called DNS amplification attack [59].

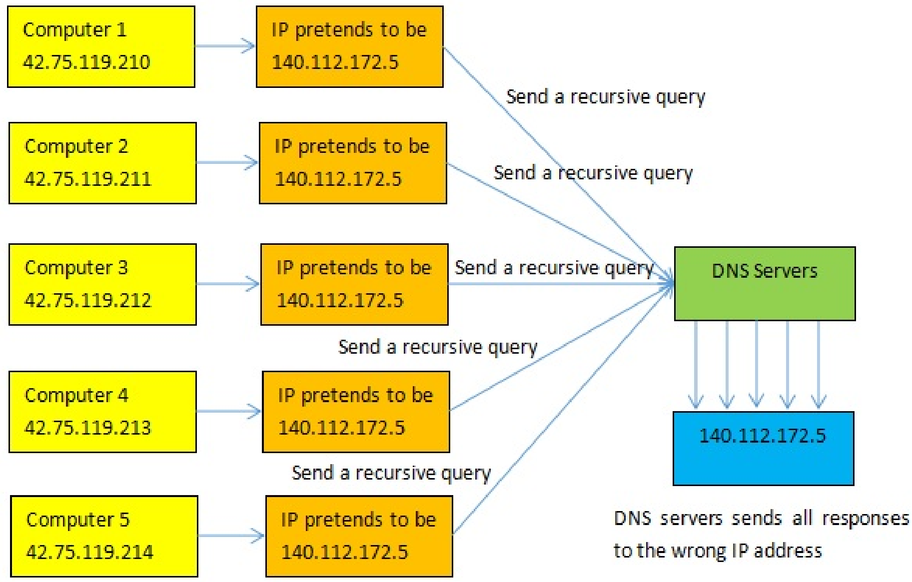


Figure 6.1: *The DDOS attack in recursive DNS queries [8]*

Thus, restricting DNS queries may be the ideal method to prevent DNS amplification attacks, the implementation is to disable the recursion for everyone, only local queries are allowed to be processed [61].

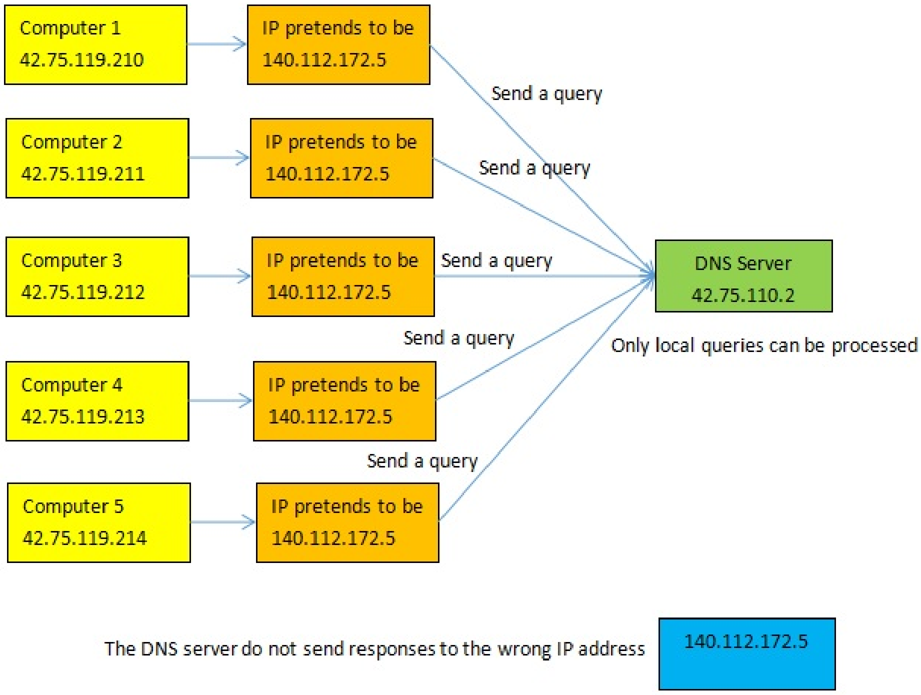


Figure 6.2: *Restricting DNS queries to prevent DNS amplification attacks [8]*

6.2 The technical plan for constructing TRR in Ireland

...

District	**Population**	**Queries P.S.**	**Minimal**	**Recommend**
Dublin city	554,554			
Fingal	296,020			
South Dublin	278,767			
Dún Laoghaire–Rathdown	218,018			
Cork	542,868			
Galway	258,058			
Kildare	222,504			
Meath	195,044			
Limerick	194,899			
Tipperary	159,553			
Donegal	159,192			
Wexford	149,722			
Kerry	147,707			
Wicklow	142,425			
Mayo	130,507			
Louth	128,884			
Clare	118,817			
Waterford	116,176			
Kilkenny	99,232			
Westmeath	88,770			
Laois	84,697			
Offaly	77,961			
Cavan	76,176			
Sligo	65,535			
Roscommon	64,544			
Monaghan	61,386			
Carlow	56,932			
Longford	40,873			
Leitrim	32,044			
Total	4,761,865			

Table 6.1: The required numbers of DOH servers in Irish counties (exclude Northern Ireland, the United Kingdom part)

Bibliography

[1] pmtk.medium.com, "Compare dnssec, dnscrypt, dns over tls, dns over https." [Online]. Available: https://pmtk.medium.com/%E5%AF%B9%E6%AF%944%E7%A7%8D%E5%BC%BA%E5%8C%96%E5%9F%9F%E5%90%8D%E5%AE%89%E5%85%A8%E7%9A%84%E5%8D%8F%E8%AE%AE-dnssec-dnscrypt-dns-over-tls-dns-over-https-2b6faca60892.

[2] root servers.org, "root-servers.org." [Online]. Available: https://root-servers.org/.

[3] Akamai.com, "Dns trends and traffic." [Online]. Available: https://www.akamai.com/de/de/why-akamai/dns-trends-and-traffic.jsp.

[4] AMS-IX, "Total traffic statistics(amsterdam)." [Online]. Available: https://stats.ams-ix.net/index.html.

[5] Cloudflare, "What is a dns root server?." [Online]. Available: https://www.cloudflare.com/learning/dns/glossary/dns-root-server/.

[6] stats.dns.icann.org, "stats.dns.icann.org." [Online]. Available: https://stats.dns.icann.org/.

[7] W. T. Hamza Boulakhrif, Yuri Schaeffer, "Analysis of dns resolver performance measurements," tech. rep., University of Amsterdam, July 2015.

[8] F. Internet, "What is recursive dns and why is it not recommended for most server owners?." [Online]. Available: https://www.youtube.com/watch?v=W3wXkHAv3qo.

[9] Mozilla, "Trusted recursive resolver." [Online]. Available: https://wiki.mozilla.org/Trusted_Recursive_Resolver.

[10] J. Peters, "What is dns, how it works + vulnerabilities." [Online]. Available: https://www.varonis.com/blog/what-is-dns/.

[11] Wikipedia, "Domain name system security extensions." [Online]. Available: https://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions.

[12] Wikipedia, "Dnscrypt." [Online]. Available: https://en.wikipedia.org/wiki/DNSCrypt.

[13] Wikipedia, "Dns over tls." [Online]. Available: https://en.wikipedia.org/wiki/DNS_over_TLS.

[14] Wikipedia, "Dns over https." [Online]. Available: https://en.wikipedia.org/wiki/DNS_over_HTTPS.

[15] Wikipedia, "Ipv4." [Online]. Available: https://en.wikipedia.org/wiki/IPv4.

[16] Wikipedia, "Ipv6." [Online]. Available: https://en.wikipedia.org/wiki/IPv6.

[17] Wikipedia, "Domain name." [Online]. Available: https://en.wikipedia.org/wiki/Domain_name.

[18] D. M. Easy, "Authoritative vs. recursive dns servers: What's the difference?." [Online]. Available: https://medium.com/@DNSMadeEasyBlog/authoritative-vs-recursive-dns-servers-whats-the-difference-d0e5821c7617.

[19] Wikipedia, "Request for comments." [Online]. Available: https://en.wikipedia.org/wiki/Request_for_Comments.

[20] Wikipedia, "Internet engineering task force." [Online]. Available: https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force.

[21] S. Bortzmeyer, "Dns privacy considerations." [Online]. Available: https://tools.ietf.org/html/rfc7626, August 2015.

[22] E. T. Wicinski, "Dns privacy considerations draft-ietf-dprive-rfc7626-bis-08." [Online]. Available: `https://tools.ietf.org/html/draft-ietf-dprive-rfc7626-bis-08`, October 2020.

[23] Wikipedia, "Public recursive name server." [Online]. Available: `https://en.wikipedia.org/wiki/Public_recursive_name_server`.

[24] T. D. Institute, "Disadvantages of dnssec." [Online]. Available: `https://dnsinstitute.com/documentation/dnssec-guide/ch06s06.html`.

[25] IONOS, "Dns over tls: an improved security concept." [Online]. Available: `https://www.ionos.com/digitalguide/server/security/dns-over-tls/`.

[26] M. Brinkmann, "Comcast is the first isp that joins firefox's trusted recursive resolver program." [Online]. Available: `https://www.ghacks.net/2020/06/26/comcast-is-the-first-isp-that-joins-firefoxs-trusted-recursive-resolver-prog`
June 2020.

[27] L. Abrams, "Mozilla enables dns-over-https by default for all usa users." [Online]. Available: `https://www.bleepingcomputer.com/news/software/mozilla-enables-dns-over-https-by-default-for-all-usa-users/`.

[28] S. Gatlan, "Firefox 77.0.1 released to prevent ddosing doh dns providers." [Online]. Available: `https://www.bleepingcomputer.com/news/security/firefox-7701-released-to-prevent-ddosing-doh-dns-providers/`.

[29] C. Cimpanu, "Google to run dns-over-https (doh) experiment in chrome." [Online]. Available: `https://www.zdnet.com/article/google-to-run-dns-over-https-doh-experiment-in-chrome/`.

[30] M. Jackson, "Google chrome joins firefox – soft defaults to dns over https." [Online]. Available: `https://www.ispreview.co.uk/index.php/2020/05/google-chrome-joins-firefox-soft-defaults-to-dns-over-https.html`.

[31] J. E. Dunn, "Chrome 83 adds dns-over-https support and privacy tweaks." [Online]. Available: `https://nakedsecurity.sophos.com/2020/05/21/chrome-83-adds-dns-over-https-support-and-privacy-tweaks/`.

[32] Winaero, "Enable dns over https in opera (doh)." [Online]. Available: https://winaero.com/enable-dns-over-https-in-opera-doh/.

[33] C. Cimpanu, "Here's how to enable doh in each browser, isps be damned." [Online]. Available: https://www.zdnet.com/article/dns-over-https-will-eventually-roll-out-in-all-major-browsers-despite-isp-op

[34] Wikipedia, "Root name server." [Online]. Available: https://en.wikipedia.org/wiki/Root name server.

[35] D. M. Easy, "What is the difference between authoritative and recursive dns nameservers?." [Online]. Available: https://umbrella.cisco.com/blog/what-is-the-difference-between-authoritative-and-recursive-dns-nameservers.

[36] central statistics office (Ireland), "Information society statistics - households." [Online]. Available: https://www.cso.ie/en/releasesandpublications/er/isshh/informationsocietystatistics-households2018/.

[37] J. Clement, "Global digital population as of july 2020." [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide/.

[38] Worldometer, "Ireland population(live)." [Online]. Available: https://www.worldometers.info/world-population/ireland-population/.

[39] N. Cumins, "Avoiding the internet rush hour." [Online]. Available: https://broadbanddeals.co.uk/avoiding-the-internet-rush-hour/.

[40] ix.br, "Selecione a localidade para ver as estatísticas de tráfego." [Online]. Available: https://ix.br/trafego/agregado/sp.

[41] federal communications commission, "Measuring broadband america-july 2012." [Online]. Available: https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-july-2012.

[42] DE-CIX, "Traffic statistics(berlin)." [Online]. Available: https://www.bcix.de/bcix/traffic/.

[43] Wikipedia, "Melbourne." [Online]. Available: https://en.wikipedia.org/wiki/Melbourne.

[44] Wikipedia, "Comparison of dns server software." [Online]. Available: https://en.wikipedia.org/wiki/Comparison of DNS server software.

[45] K. John, "Bind vs dnsmasq vs powerdns vs unbound." [Online]. Available: `https://computingforgeeks.com/bind-vs-dnsmasq-vs-powerdns-vs-unbound/`.

[46] ISHM, "Building dns over https server on centos 7." [Online]. Available: https://ishm.idv.tw/?p=481.

[47] Archlinux, "Dns over https servers." [Online]. Available: `https://wiki.archlinux.org/index.php/DNS_over_HTTPS_servers`.

[48] TINYDNS, "Compare the different dns servers: Which one is right for you?." [Online]. Available: `https://tinydns.org/compare-different-dns-servers/`.

[49] Wikipedia, "Nsd." [Online]. Available: https://en.wikipedia.org/wiki/NSD.

[50] Wikipedia, "Unbound (dns server)." [Online]. Available: `https://en.wikipedia.org/wiki/Unbound(DNS_server)`.

[51] Wikipedia, "Bind." [Online]. Available: https://en.wikipedia.org/wiki/BIND.

[52] Wikipedia, "Powerdns." [Online]. Available: https://en.wikipedia.org/wiki/PowerDNS.

[53] Loull, "Dns resources." [Online]. Available: https://www.cnblogs.com/549294286/p/5200255.

[54] Facebookexperimental, "Dns over https proxy." [Online]. Available: `https://github.com/facebookexperimental/doh-proxy`.

[55] NGINX, "Welcome to niginx wiki!." [Online]. Available: https://www.nginx.com/resources/wiki/.

[56] Wikipedia, "List of linux distributions." [Online]. Available: `https://en.wikipedia.org/wiki/List_of_Linux_distributions`.

[57] Sciencebuddies.org, "Variables in your science fair project." [Online]. Available: `https://www.sciencebuddies.org/science-fair-projects/science-fair/variables`.

[58] D. Geek, "What is nxdomain?." [Online]. Available: `https://www.dnsknowledge.com/whatis/nxdomain-non-existent-domain-2/`.

[59] Imperva, "Dns amplification." [Online]. Available: `https://www.imperva.com/learn/ddos/dns-amplification/`.

[60] Cloudflare, "What is recursive dns?." [Online]. Available: `https://www.cloudflare.com/learning/dns/what-is-recursive-dns/`.

[61] F. Internet, "What is recursive dns and why is it not recommended?." [Online]. Available: `https://help.fasthosts.co.uk/app/answers/detail/a_id/1276`.

Appendix

...