**Project Title:** Domain Name System, a comparative study.

Author

Oisin Michael Farrell. <[oisinf@kth.se](mailto:oisinf@kth.se)>

Abstract

In this report I performed a series of experiments to see if the default Domain Name System (DNS) gave better query times than some other available DNS servers. I did this to better my understanding of the DNS. My experiment showed that the default DNS did give smaller query times for certain domain names but not for all. Therefore, I would have to conduct a more thorough experiment to conclude concretely that the default server giver better query times overall.

# Introduction

The aim of this research paper was to get a better and more comprehensive understanding of the Domain Name System (DNS). To form this understanding, I want to dissect the inner workings of the DNS with some experiments that will examine the performance of my own default DNS and some other popular DNS servers. For this paper, I performed a comparative study between the default DNS and some common ones used in internetworking today. The reason that I have chosen this topic to study is that DNS was totally new to me when I started this course and I wanted to better familiarise myself with this topic.

## Theoretical framework/literature study

As we know, each host and website are equipped with an IP address [1]. Computers are not the only devices that have IP addresses associated with them. Web servers also have an IP address [11]. The problem is that it is impossible to remember the IP address of all your favourite sites. A DNS server is somewhat like a phone book. You can look up a person’s name and find their corresponding phone number. In a similar manner you do not need to know the IP address of the site you want to access but instead you just need to know the hostname for example “www.facebook.com” and the DNS will take care of the rest.

A DNS is a system that is used to translate host names into IP addresses [12] just like the phone book gives you the number associated with a particular person. There are two problems with implementing a DNS like a phone book.

1. There are far too many hosts to simply keep a record of them and look up each time we want to reach them [7].
2. IP addresses can be dynamic [13] and so will constantly change meaning there would be no use to have a static record of the host and their IP address. Trying to dynamically update an entire ‘phone book’ of IP addresses would take to much time.

Instead, DNS is built in a tree like hierarchical structure [5, 7]. At the top of the tree are the root servers. There are 13 of these root servers all around the world [8] and each of these contains information that makes up the root zones. These root zones are global lists of Top-Level Domains (TLD) [17]. Each TLD then takes care of a different domain name suffix such as “.com” or “.org” [3].

Graphical user interface, application

Description automatically generated

Figure 1: The hierarchical Structure of a DNS. Diagram inspired by [6].

A domain is a node in the DNS tree [6]. Domains in the TLD section represent organizations or entities and these can be further split into subdomains [6]. For example, in Figure 1, we have the google subdomain under the .com TLD and this domain is further divided into the mail.google subdomain. This makes it easier to query a DNS server. When we send a DNS query to find us the IP address of a certain website, for example linux.org, the resolver first checks to see if the website that we are looking for in our DNS cache. A DNS cache is a record of recent queries. This reduces query times as the DNS resolver does not have to go looking for the IP address of the website as it has it stored already [14]. If the DNS does not have an entry for that query, then the query is sent to the resolver. The root server passes the query on to the correct TLD depending on the websites suffixes such as .com or .net. Each TLD stores information about the second level domain for a given suffix. The TLD will give the location of the name server for a particular website and here the IP address of the website is stored.

During my research for this paper, I came across the Time to Live (TTL) value of a DNS query. The TTL value is the length of time in milliseconds that a DNS will keep a record of a query result [15]. The TTL is based on certain factors such as how tolerant a user is to out of date data and if the data is static or not [10]. TTL time varies from roughly 5 minutes (300 seconds) to 24 hours (86400) which are typical TTL times [16].

## Research questions.

The main research question of this paper is whether the default DNS gives faster query times than other DNS servers. Aside from this I also wanted to look to see do different DNS servers give different TTL values.

# Method

The main experiment of this paper was to find out if the default DNS server had a better query time than the other DNS servers. To do this experiment I would query my default DNS server and three other DNS servers and compare their performance. In the interest of making the experiment fair there were two considerations taken before I ran the experiment. The first consideration was that the default DNS cache should be flushed after every query to eliminate bias as once a query is made the address is stored and all subsequent queries will produce results much faster. To show this I performed a small experiment where I query the DNS server three times without flushing the cache and then three times after flushing the cache, showing the effect of the cache.

The second consideration was to query a variety of websites. I did this because DNS servers themselves have their own cache. This means that if I query two DNS servers one might return the query to a particular website faster than the other not because it is in fact a faster DNS server but because it already had an entry for that website. To avoid this bias, I will query three websites in my experiment, one of which will be a site will low traffic so that hopefully none of the DNS servers will have an entry for it in their cache.

The experiment to check TTL values of the DNS queries has an identical set up to the experiment for checking query time except that TTL is recorded rather than query time.

# Results and Analysis

Table contains the results from a small experiment and these results show effect the cache has on DNS query time.

Table 1, Results of cache experiment.

|  |  |  |
| --- | --- | --- |
| DNS Server | Query time without flushing cache | Query time with flushing cache |
| Default (127.0.0.53) | 70, 4, 4 ms | 96, 96, 104 ms |
| Google (8.8.8.8) | 132, 112, 148 ms | 136, 160, 72 ms |
| Cloudflare (1.1.1.1) | 64, 84, 140 ms | 108, 340, 132 ms |
| Neustar (64.6.64.6) | 140, 100, 92 ms | 156, 124, 172 ms |

From the table it is evident that the cache influences the query time for the default DNS server. As you can see, in the default DNS servers’ case, the first query time is 70ms and the following two have been reduced to 4ms. This is because once the query is returned the DNS will cache the IP address and for any other queries to the same webpage the resolver will simply return this IP address without having to make a request to any other servers thus decreasing the query time. This is trivial and I am sure that the reader already knew how the cache works; however, one interesting finding from this experiment was that when querying a DNS server that is not the default, such as Google’s 8.8.8.8 server, the flushing of the cache has no effect which was interesting to me.

Table 2 shows the results of the experiment where I compared the TTL value for the four DNS servers.

Table 2: TTL times for various DNS servers

|  |  |  |
| --- | --- | --- |
| Website | | |
| Time to Live.  (Seconds) | Linux.org | Wikipedia.org | Playlistmachinery.com |
| DNS Server | Default (127.0.0.53) | 300, 292, 287 | 141, 139, 137 | 84631, 84589, 84544 |
| Google  (8.8.8.8) | 299, 291, 299 | 162, 141, 146 | 20769, 26759, 21599 |
| Cloudflare  (1.1.1.1) | 221, 218, 208 | 382, 377, 365 | 85430, 85410, 85403 |
| Neustar  (64.6.64.6) | 300, 300, 300 | 237, 471, 226 | 43200, 43200, 43193 |

As you can see the TTL value does not vary so much for the common websites (linux.org and Wikipedia.org) across all DNS servers, including the default server. However, in the case of the website with low traffic the TTL varies quite dramatically. We see that the default server and Cloudflare’s give a TTL value of roughly 24 hours while Neustar gives about 12hours. Google’s server had the lowest TTL with around roughly 6 hours. It is evident from the data above that the TTL varies depending on the server. While conducting the experiment to obtain the results above the DNS cache was flushed each time a query was done. I did this to keep consistency throughout my experiment. One interesting thing that I noticed in the case of the default DNS server was that the TTL value was influenced by the DNS cache.

Table 3, TTL experiment without flushing the cache.

|  |  |
| --- | --- |
| Time to Live.  (Seconds) | Website:  Playlistmachinery.com |
| DNS Server | Default (127.0.0.53) | 84329, 7194, 7193 |

As you can see, not flushing the cache had a profound effect on the TTL value for the default server. This makes sense as the TTL value represents the time that can elapses before a refreshed version of the data needs to be recollected. If a DNS query is sent for a particular website, is it likely that the user will access this website again and it therefore makes sense to update the information at more frequent intervals. You will notice here that the other DNS servers are not present and there is only data from the low traffic website. This is because flushing the default cache will not influence the cache of the other servers. Data for the other websites are not here as not flushing the cache does not have much of an effect on the TTL time as they were already quite low.

Table 4 shows the results of the main experiment. Each query was run three times and then the median of the three query times was calculated and documented. This was done to get rid of any one-time values that might skew the results.

Table 4, Query response time experiment.

|  |  |  |
| --- | --- | --- |
| Website | | |
|  | Linux.org | Wikipedia.org | Playlistmachinery.com |
| DNS Server | Default (127.0.0.53) | 68, 60, 120  Median: 60ms. | 108, 164, 160  Median: 160ms | 170, 220, 92ms  Median: 170ms |
| Google  (8.8.8.8) | 152, 284, 92  Median: 152ms | 84, 56, 152  Median: 84ms | 180, 288, 280ms  Median: 280ms |
| Cloudflare  (1.1.1.1) | 112, 120, 88  Median: 112ms | 104, 124, 312  Median: 124ms | 384, 176, 152  Median: 176ms |
| Neustar  (64.6.64.6) | 136, 136, 136  Median: 136 ms | 140, 132, 136  Median: 132 ms | 240, 236, 236  Median: 236ms |

As you can see from the results different DNS resolvers give very different query times. The default had the quickest query time for the linux.org and playlistmachiebery.com cases. However, it had the second slowest for the Wikipedia.org case so it is hard to conclude that it is the fast DNS server with absolute certainty.

# Discussion

The research question presented in this report was to find out if the default servers gave better response times than other DNS servers. Having carried out the experiment I cannot concretely say that the default DNS server gives better response times across the board. It did give better response time for certain websites, but I would need to do a more thorough study before I made any explicit conclusions.

From doing this report I learned many things that I did not explicitly set out to find. The first of which is that each DNS server has their own cache. This is evident from the fact that each DNS server’s slowest query was the website thought to have the least traffic. I now know that the reason for this is that popular websites such as Wikipedia.org are stored in the cache of the DNS server itself. I assumed that only the local machine had a cache. Regarding the TTL value I learned that TTL depends on how frequently a domain name is lookup up, hence a web site is visited frequently, then it is likely different DNS servers have different standards for what TTL they will use.

Overall, I think this was a successful research paper. If I had more time, I would have looked at more DNS servers and queried them for a larger variety of websites. Also, I would have liked to set up my own DNS server and compare its performance to the currently available DNS servers.

References

[1] - James F. Kurose, Computer Networking: A Top-Down Approach: International Edition 2013, Pearson Education UK, Available from: ProQuest Ebook Central. [12 April 2021].

[2] - P. Mockapetris and K. J. Dunlap. 1988. Development of the domain name system. SIGCOMM Comput. Commun. Rev. 18, 4 (August 1988), 123–133. DOI:https://doi.org/10.1145/52325.52338

[3] - Techquickie, (20 May 2016), DNS as Fast as Possible,

[online video] Available at: <https://www.youtube.com/watch?v=Rck3BALhI5c&t=276s>

Accessed: 12th April 2021

[4] - Computerphile, (9 Jul 2020), How DNS works, [online video], Available at:

<https://www.youtube.com/watch?v=uOfonONtIuk>,

Accessed: 12th April 2021

[5] P. Mockapetris and K. J. Dunlap, ‘Development of the Domain Name System’, SIGCOMM Comput. Commun. Rev., vol. 18, no. 4, pp. 123–133, Aug. 1988 [Online]. DOI: 10.1145/52325.52338

[6] – Novell DNS/DHCP Management. “DNS Hierarchy.” Novell. <https://www.novell.com/documentation/dns_dhcp/?page=/documentation/dns_dhcp/dhcp_enu/data/behdbhhj.html> (Accessed: 13th April 2021). `

[7] - P. V. Mockapetris, ‘Domain names - concepts and facilities’, Internet Request for Comments, vol. RFC 1034 (INTERNET STANDARD), Nov. 1987 [Online]. DOI: 10.17487/RFC1034

[8] – IONOS Inc., ‘What is a root server?, May 8, 2019, Definition and mechanics’, ionos.com. <https://www.ionos.com/digitalguide/server/know-how/what-is-a-root-server-definition-and-background/#:~:text=In%20total%2C%20there%20are%2013,for%20Assigned%20Names%20and%20Numbers>). (accessed April 13, 2021)

[9] – VeriSign, Inc., ‘How the Domain Name System (DNS) works’, verisign.com. <https://www.verisign.com/en_US/website-presence/online/how-dns-works/index.xhtml> (accessed April 13, 2021)

[10] – M. Brinkley and J. Chhabra, “Caching challenges and strategies”, awsstatic.com. <https://d1.awsstatic.com/builderslibrary/pdfs/caching-challenges-and-strategies.pdf?did=ba_card-body&trk=ba_card-body> (accessed April 15, 2021)

[11] – K. Wood, “What is an IP address?”, September 13th 2019 , hostgator.com. <https://www.hostgator.com/blog/what-is-ip-address/#:~:text=Your%20server%20has%20an%20IP,to%20the%20internet%20can%20change>. (accessed April 15, 2021)

[12] – Cloudflare, “What is DNS?”, cloudflare.com. <https://www.cloudflare.com/learning/dns/what-is-dns/?utm_referrer=https://www.cloudflare.com/learning/dns/what-is-dns/> (accessed April 15, 2021)

[13] – Eltoro, “How Long Does an IP Address Stay Attached to a Home or Business?”, eltoro.com. <https://eltoro.com/how-long-does-an-ip-address-stay-attached-to-a-home-or-business/#:~:text=Or%20put%20more%20simply%20DHCP,to%20your%20home%20or%20business.&text=Your%20ISP%20can%20either%20configure,dynamic%20(it%20can%20change).&text=Most%20residential%20customers%20have%20a,meaning%20that%20it%20can%20change>. (accessed April 15, 2021)

[14] – Cloudflare, “What is caching?”, cloudflare.com. <https://www.cloudflare.com/learning/cdn/what-is-caching/> (accessed April 15, 2021)

[15] – NS1, “Understanding TTL values in DNS records”, ns1.com. <https://ns1.com/resources/understanding-ttl-values-in-dns-records> (accessed April 16, 2021)

[16] – IONOS Inc, “Understanding and configuring DNS TTL”, ionos.com. <https://www.ionos.com/digitalguide/server/configuration/understanding-and-configuring-dns-ttl/#:~:text=TTL%20is%20given%20in%20seconds,DNS%20changes%20to%20take%20effect>. (accessed April 16, 2021)

[17] – Netnod, “What are root name servers?”, Netnod.se.

<https://www.netnod.se/i-root/what-are-root-name-servers#:~:text=Root%20name%20servers%20are%20the,numeric%20addresses%20such%20as%20192.71>. (accessed April 19, 2021)