Bachelor Informatica

University of Amsterdam

Dissecting the top-level domains in the domain name system

Constantijn Bicker Caarten

June 9, 2017

Supervisor(s): ir. R. Dolmans (NLnet Labs) & dr. ir. A. Taal (UvA)

Signed:

Abstract

This thesis presents a framework that is used to monitor the quality of TLDs in the DNS. The properties it measures are credibility, performance and robustness. It determines the quality of credibility by analysing the use of DNSSEC, the quality of performance by measuring response time and the quality of robustness by analysing network diversity, reachability and name servers. The results of these properties are then split into groups to see which of these groups of TLDs are preforming well or poorly. The groups that are compared in this thesis are ccTLDs versus gTLDs and old TLDs versus new TLDs. In this thesis new TLDs are defined as TLDs that are created since the introduction of the new gTLD program which was created with the purpose to enhance competition, innovation and consumer choice. The results reflect that this purpose has been successful in introducing new TLDs of good quality to the DNS and that some of the old TLDs are not keeping up with the current state of the art.

Contents

CHAPTER 1

Introduction

1.1 Problem definition

In the past few years the number of top-level domains (TLDs) in the Domain Name System (DNS) has rapidly increased. As of writing this thesis 1531 TLDs are listed in the root zone. This can be seen in figure 1.1 which uses data that is obtained through the WHOIS service of the Internet Corporation for Assigned Names and Numbers (ICANN). The vast majority of the current set of TLDs are generic TLDs (gTLDs) with various purposes which are mostly commercial. It is important that these new TLDs uphold a level of quality as disruption can cause financial and societal damage. In this thesis a framework will be described that is used to monitor the quality of TLDs through certain properties to see if these new TLDs uphold a level of quality or if the older TLDs need improvement because of the current state of the art in the DNS.

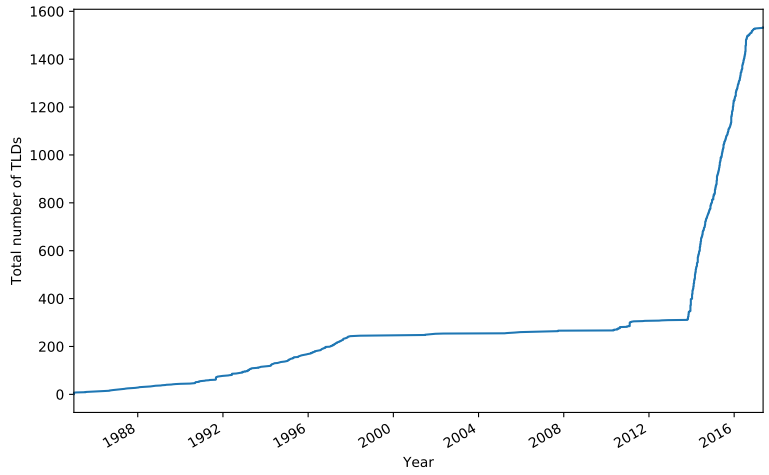

Figure 1.1: Number of TLDs over time.

The quality of TLDs will be monitored by gathering data on credibility, performance and robustness. For credibility the use of DNSSEC will be analysed. Performance is determined by measuring the response time of TLD name servers. Robustness is divided into several sub properties. One of these sub properties is network diversity which is determined by looking how many networks a TLD is hosted in and how many TLDs a network hosts. Furthermore,

robustness is measured by investigating the number of name servers per TLD and determining the support of the protocols TCP and UDP over IPv4 and IPv6 on name servers.

1.2 Related work

Most related work on determining measuring properties of TLDs focuses on bench marking the performance of name servers. Part of these benchmarks are done internally[2]. As this is impossible to do for all name servers of each TLD, this thesis will measure performance externally. Measurements done from the outside have been done as well, but the focus is on a sub selection of TLDs or the root[3, 10]. The performance measured in this thesis expands on this research by including every TLD in the DNS.

Most research on the credibility of a TLD is done through psychological experimentation that for example show that end users are more likely to trust the content of web page that has a `.gov` TLD than the content of web page with a `.com` TLD[14]. There also exists research on credibility that is more technical but those focus solely on country-code TLDs (ccTLDs), TLDs reserved for countries, or only look at the adoption rate within each TLD zone[1]. In this thesis this data is expanded on by dissecting the credibility of all TLDs by looking at which TLDs support DNSSEC and are actually using it and which algorithms they are using.

The robustness of a TLD is determined by looking at network diversity, number of name servers, TCP/UDP support over IPv4 and IPv6 and the organisations behind them. Part of this data is available online, but this thesis will provide an up to date version of this data and more insight by looking at the data by splitting them between older and newer TLDs.

1.3 Research definition

The main purpose of this thesis is to determine the quality of every TLD by measuring properties such as credibility, performance and robustness. In order to accomplish this a framework is proposed that monitors these properties by gathering data. Then TLDs are grouped by gTLDs and ccTLDs to see if there is a difference in the results between these two groups. Additionally, the same is done by differentiating between older and newer TLDs. These groups are made by splitting TLDs created before and after 2013, because this is the year when the number of TLDs rapidly increased as shown in figure 1.1. This rapid increase is due to the New gTLD Program from ICANN that allowed for new gTLDs to be created in order to enhance competition, innovation and consumer choice[9]. Therefore, this thesis differentiates between TLDs created before and after 2013 as old and new TLDs.

[1] `https://powerdns.org/dnssec-stats/`

CHAPTER 2

Theoretical background

2.1 DNSSEC

Domain Name System Security Extensions (DNSSEC) is an extensions to DNS that ensures data integrity. DNSSEC makes sure that the response contains the correct data. It does this by using a chain of trust. This chain of trust following the same order as the DNS hierarchy as can be seen in figure 2.1. This chain of trust works by signing records by using public-key cryptography. The type of algorithms that can be used are listed with their numerical identifiers by Internet Assigned Numbers Authority[1] (IANA), but the standard only lists the most used algorithms shown in table 2.1. These algorithms are used by the key signing key (KSK) and the zone signing key (ZSK). The ZSK is used to sign all the records of the zone except for `DNSKEY`. The KSK is used to sign the `DNSKEY`s and its hash is used to create the `DS` record and placed in the parent. The DS record of the root zone is put in the validating resolver which works as the trust anchor. The trust anchor is used as a beginning point for the data validation. Then when a DNS look up is performed the `DS` and `DNSKEY` records are used to validate the data for each name server response[4].

Must Implement	Must Not Implement	Recommended to Implement	Optional
RSASHA1	RSAMD5	RSASHA256 RSASHA1-NSEC3-SHA1 RSASHA512 ECDSAP256SHA256 ECDSAP384SHA384	Any registered algorithm not listed in this table

Table 2.1: DNSSEC Implementation Status[12].

2.2 Autonomous system

An Autonomous System (AS) is network of one or more IP prefixes owned by a single administrative entity. An AS is operated by one ore more network operators. Each AS has its own routing policy[8]. Each AS has a numerical representation that is used to identify an AS called the Autonomous System Number (ASN). This number is a 32-bit number meaning it ranges from 0 to 4294967295[6].

[1] `https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml#dns-sec-alg-numbers-1`

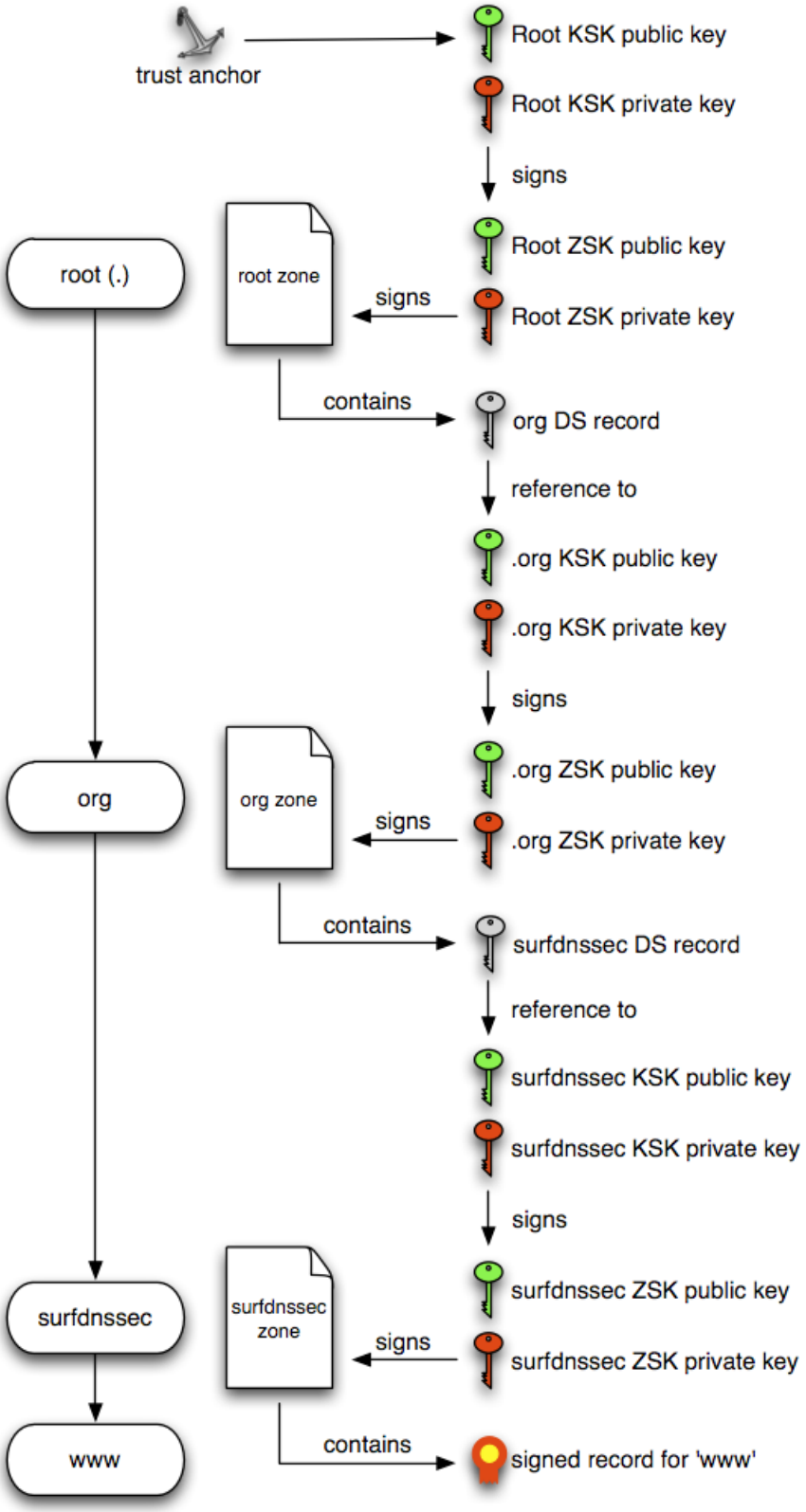

Figure 2.1: Chain of trust required to validate the answer to a query for www.surfdnssec.org (Source: Surfnet)

2.3 Anycast

Anycast is an addressing method that makes use of hosting an IP prefix on different networks. A query for an IP from such an IP prefix gets routed to one of the different networks. By using anycast this routing will ensure that it will go to the nearest available network. In the DNS this can be used to host the same zone on several different networks spread around the world[7]. This has multiple benefits such as load balance and faster response time.

CHAPTER 3

Implementation

3.1 Data collection

3.1.1 Jupyter Notebook

All the data discussed throughout this thesis will be collected using python scripts run in Jupyter Notebook[1]. Jupyter Notebook is a web application in which documents with code can be created. By using Jupyter Notebook the data remains loaded in the memory which makes it easier to manipulate. It is also easier to validate in between results as code can be run piece by piece by spreading it over different cells. Another advantage is that shell commands can be run directly in a shell and their output can be saved as Python variables. This in combination with the vast amount of Python packages will provide a foundation in which all desired results can be obtained through.

3.1.2 dig

As mentioned, Jupyter Notebook supports shell commands to be run along side Python code. One frequently used shell command for DNS data retrieval is `dig` as it is able to gather various information such as name servers and IP addresses . This command is mostly run with the query options `+noall` and `+answer` as this will only return the answer section from the response making it easier to parse it in Python. The query option `noidn` is also included to make parsing easier because if internationalised domain name (IDN) is enabled the column order changes with languages that are read from right to left.

3.1.3 RIPE Atlas

Certain measurement results might vary depending on the location they are performed from. In order to avoid such a location bias, RIPE Atlas[2] is used as a measurement tool. By using RIPE Atlas various hardware devices (probes) spread around the world can be used to perform measurements such as ping, traceroute, DNS, SSL/TLS, NTP or HTTP. Although these probes are spread around the globe, they are more densely located in the regions Europe and North America than in other regions and as such using all probes would still maintain a bias. To work around this an equal amount of probes are selected from the 5 regions defined by RIPE Atlas as shown in figure 3.1.

[1] `https://jupyter.org/`
[2] `https://atlas.ripe.net/`

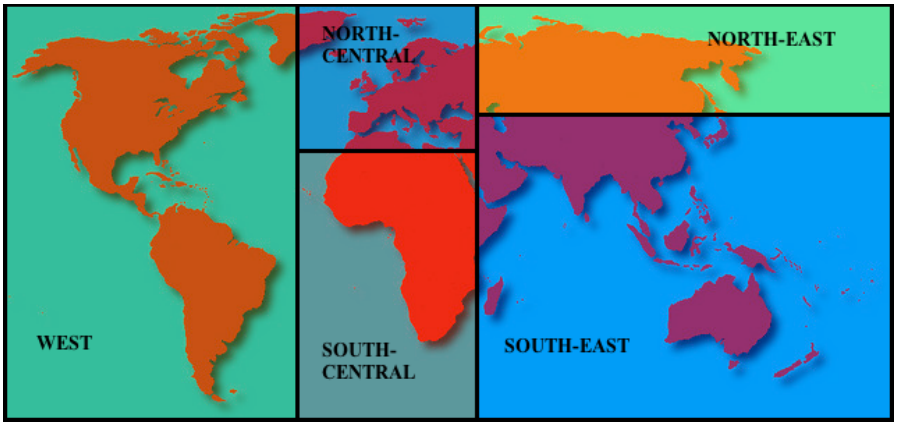

Figure 3.1: RIPE Atlas regions

3.1.4 Cymru

The organisation Team Cymru Research NFP[3] has various initiatives related to Internet measurements. One of these initiatives is their *IP to ASN Mapping* and will be used to map IP addresses to ASN as the name suggests. Using this service the following information can be obtained from an IP address:

- BGP Origin ASN
- BGP Peer ASN
- BGP Prefix
- Prefix Country Code (assigned)
- Prefix Registry (assigned)
- Prefix Allocation date
- ASN Country Code (assigned)
- ASN Registry (assigned)
- ASN Allocation date
- ASN Description

```
$ whois -h whois.cymru.com " -v 193.176.144.5"
AS      | IP               | BGP Prefix          | CC | Registry | Allocated  | AS Name
1140    | 193.176.144.5    | 193.176.144.0/24    | NL | ripencc  |            | SIDN, NL
```

It is also possible to do bulk operation by providing a list of IP addresses. The file containing the list of IP addresses each separated with a newline. Also the first line of the file must the word *begin* and the last line must be the word *end*. Optionally the second line of the list can be set to *verbose* in order to obtain every detail provided by Cymru.

```
begin
verbose
196.216.168.54 d.nic.so.
204.61.216.101 e.nic.so.
end
```

Then by using `netcat` on the WHOIS service of Cymru and providing the file with IP addresses a list of IP to ASN mappings can be obtained as follows:

[3] `https://www.team-cymru.org/`

```
$ netcat whois.cymru.com 43 < ip_addresses_file
Bulk mode;  [2017-06-09 12:29:58 +0000]
37177 | 196.216.168.54 | 196.216.168.0/24 | ZA | afrinic | d.nic.so. | AFRINIC-ANYCAST, MU
42    | 204.61.216.101 | 204.61.216.0/23  | US | arin    | e.nic.so. | WOODYNET-1 - WoodyNet,
```

3.2 Data analysis

3.2.1 Pandas

Pandas[4] is used to analyse data because it has all the features needed for analysing the data. One of the useful features from Pandas is grouping, because with this data from name servers can be grouped by TLD. It can also filter out results, visualise data and export it to various output such as csv or latex.

[4] http://pandas.pydata.org/

CHAPTER 4

Method

The data set needed to begin gathering new data only needs to consists out of a lists of TLDs. This list can be obtained through various means like from the root zone, but this contains a lot more information than just the TLDs and so its easier to use the text file containing all TLDs from IANA[1].

4.1 Credibility

One of the factors that determines that credibility of a TLD is the use of DNSSEC because it ensures that the response given contains the correct data. Whether a TLD implements DNSSEC can be derived from a request for the type `DNSKEY` to the TLD name server and the type `DS` to the root server with `dig`. To speed up this process for all TLDs, a bulk request is performed by including a file, using the `f` flag, named `tlds` which contains all the TLDs.

```
dig +noall +answer +noidn DNSKEY -f tlds > tld_dnskeys
```

This command in Jupyter writes the `DNSKEY` answer section of the response of every TLD in the `tlds` file as a new line to the file `tld_dnskeys`. Then each line can be read and split on spaces to obtain a list which in order consists out of TLD, TTL, class, type, key type, protocol identifier, algorithm identifier and public key. Then the key type and algorithm identifier are extracted and used to determine the key type which is the key signing key if it is 257 and the zone signing key if it is 256 and the name of the algorithm by matching it with its numerical identifier. Then the `DS` record for every TLD is looked up as well to determine if they use DNSSEC similarily to the query for the `DNSKEY` but changing the type to `DS`.

```
dig +noall +answer +noidn DS -f tlds > tld_dss
```

4.2 Performance

The method used to measure performance is based on the `NXDOMAIN`-Query Technique[10]. This method makes use of a `NXDOMAIN` response from a name server to measure the response time from the name server, because a `NXDOMAIN` response ensures no further communication beyond the TLD name server is performed. In order to guarantee a `NXDOMAIN` response, a random domain is generated using the `uuid` package in Python. On the off chance this domain actually exists, a test is done to see if returns `NXDOMAIN` before hand.

Response time varies from location, so to avoid this location bias the response time is measured using RIPE Atlas with 15 probes per region. To further ensure proper measurement of response time, the name server of a TLD is cached first before measuring its response time. This caching is done by performing a DNS measurement with the query type `NS` in RIPE Atlas

[1] `https://data.iana.org/TLD/tlds-alpha-by-domain.txt`

for each TLD as query argument. The response time is then measured by performing a DNS measurement with argument type Start of Authority (`SOA`) in RIPE Atlas. These measurements are spread out over several payloads to comply with the rule of RIPE Atlas that states only 100 measurement may be run simultaneously.

It is important to note that probe resolvers are used so that the results are more consistent, because the response time from an external resolver varies per probe. Also note that the response time of TLDs are measured and not the response time of the TLD name servers, because this take accounts how well a TLD has made its name servers reachable. Measuring each name server individually would just be pointless as end users use name servers depending on their location.

4.3 Robustness

4.3.1 Name servers

One attribute that contributes to the robustness of a TLD is the number of name servers it has. This is because more name servers decreases the chance that a TLD becomes unreachable. These TLD name servers can be found by sending a DNS request for the type (`t` flag) `NS` and using the file (`f` flag) `tlds` containing all TLDs with `dig` as follows:

```
dig +noall +answer +noidn -t NS -f tlds
```

For the TLDs IN, MD, MG, MR and MX the name servers have to be gathered without bulk because these TLD names match argument values and as a result will not work in bulk mode. This is done for example for the TLD MX as follows:

```
dig +noall +answer +noidn -t NS MX
```

However, it is important to note that the number of servers returned by this request does not match up with the actual number of name servers due to anycasting. This means that the actual number of name servers for certain TLDs might be higher than is shown through DNS records.

4.3.2 Network diversity

In this thesis the number of TLDs per AS and the number ASs per TLD is what is understood as network diversity. This gives insight into how TLDs are spread over networks. It is desirable that each TLD is spread out over various networks so that if one network fails the TLD is still reachable on other networks. It is desired as well that these TLDs are spread over as much different networks, because if all TLDs are spread over the same networks then in the case those few networks were to stop working all of the TLDs would be unreachable.

The network diversity is measured by mapping the name servers to IPs and writing them to a file named `ips`. Then using Cymru these IPs are mapped to ASNs and written to a file named `asns` sorted by ASN as follows:

```
netcat whois.cymru.com 43 < ips | sort -n > asns
```

4.3.3 Reachability

In order to determine the reachability of a name server, a DNS request is send over TCP and UDP asking for the `SOA` record. This is done for both the IPv4 address and IPv6 address of each name server. If a name server does not have an IPv4 or an IPv6 address, it will skip that version of IP. Then if the name servers returns an answer it supports the protocol and if it does not return an answer it does not support the protocol. This is done using the `dnspython` library in Python as follows:

```
for datum in data:
    for p in (udp, tcp):
        # Create SOA query
        m = dns.message.make_query(datum['tld'], dns.rdatatype.SOA)
        try:
            a = p(m, datum['ip'], timeout=5)
            # We expect NOERROR RCODE (0) and an answer
            if a.rcode() == 0 and len(a.answer) > 0:
                datum[p.__name__] = True

            else:
                raise CustomDNSException('failed')
        except (dns.exception.Timeout, socket_error, CustomDNSException):
            datum[p.__name__] = False
```

4.3.4 Anycasted networks

Determining whether a name server is anycasted or not can be done by using a few probes. These probes are then spread equally distanced from one another around the world. Then each probe queries the same name server and measures the response time. If a name server is anycasted, all the response times should reflect that by being within a window of each other. For example if a probe in the Netherlands and Australia both have a response time of less than 50 ms to the same name server it is very probable that this name server is anycasted.

CHAPTER 5

Results

5.1 Credibility

The pie chart in figure 5.1 shows the distribution of algorithms used for DNSSEC of the TLDs that returned an answer asking for the DNSSKEY. The four algorithms used are RSASHA1, RSASHA1-NSEC3-SHA, RSASHA256 and RSASHA512. The most used algorithm by far is RSASHA256 followed by RSASHA1-NSEC3-SHA. RSASHA1 and RSASHA512 both are used rarely. The figure shows the results for both ZSK and KSK as they are exactly the same.

Figures 5.2, 5.3, 5.4, 5.5 and 5.6 show how many TLDs or subsets of TLDs have a `DNSKEY` record, a `DS` record or both. The vast majority of TLDs have both records. This majority is created by gTLDs because all of them have both records. New TLDs almost all have both records. Old TLDs and ccTLDs both have a somewhat equal amount of TLDs that have no records or both.

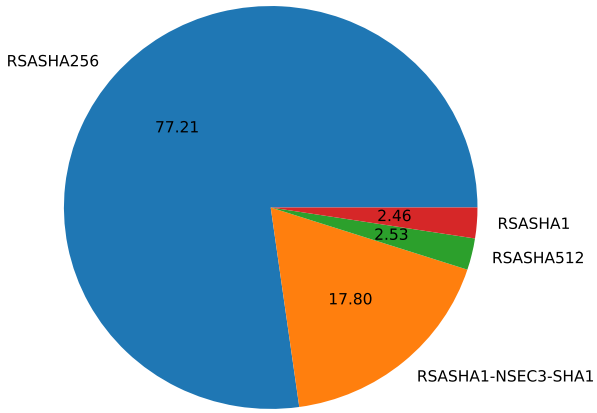

Figure 5.1: DNSSEC algorithms in use by TLDs.

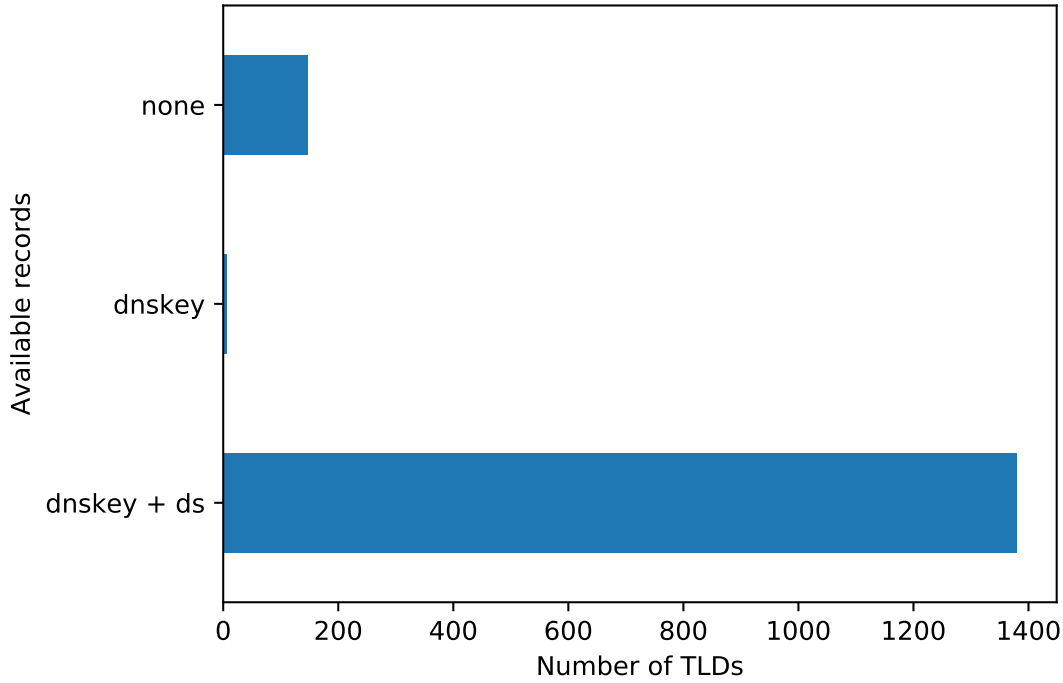

Figure 5.2: DNSSEC records of all TLDs.

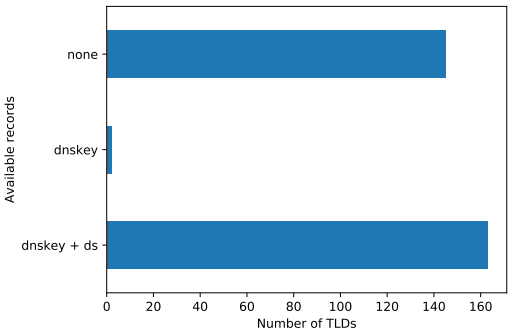

Figure 5.3: DNSSEC records of old TLDs.

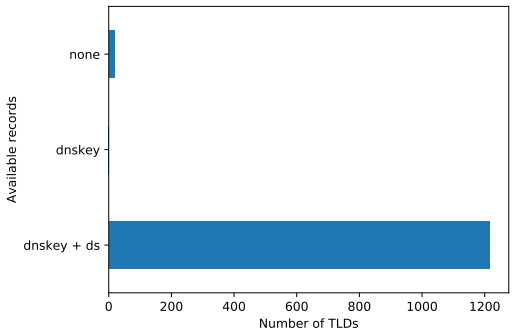

Figure 5.4: DNSSEC records of new TLDs.

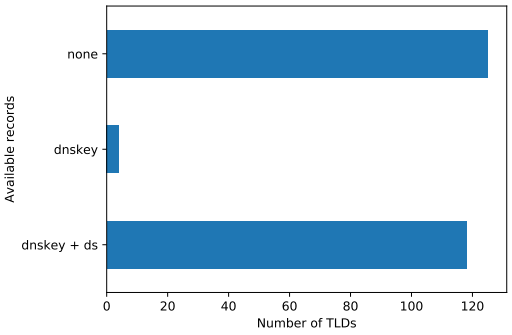

Figure 5.5: DNSSEC records of ccTLDs.

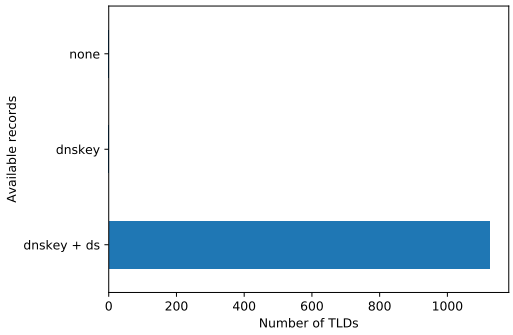

Figure 5.6: DNSSEC records of gTLDs.

5.2 Performance

Response times are measured by taking the average response time from all the probes combined for each TLD. From each region, shown in figure 3.1, 15 probes are taken which makes a total of 75 probes. The measurements are spread over three days because of the limit on how much credits can be spend on one day by RIPE Atlas[1]. Also during the measurements one to three probes became unavailable for some TLDs.

Some of the measurements resulted in time outs. These time outs occurred when it took more than 5 seconds for reply. For each TLD the number of time outs are counted as shown in figure 5.12. The amount of time outs for a TLD ranges from 0 to 7 of which 0 is the most frequent.

Figures 5.7, 5.8, 5.9, 5.10 and 5.11 are histograms that illustrates the number of TLDs for response times between 0 ms and 1400 ms. In every histogram the most frequent response time occurs around 200 ms and generally gradually lowers when the response time decreases or increases. The number of TLDs decreases faster when the response time decreases than when it increases.

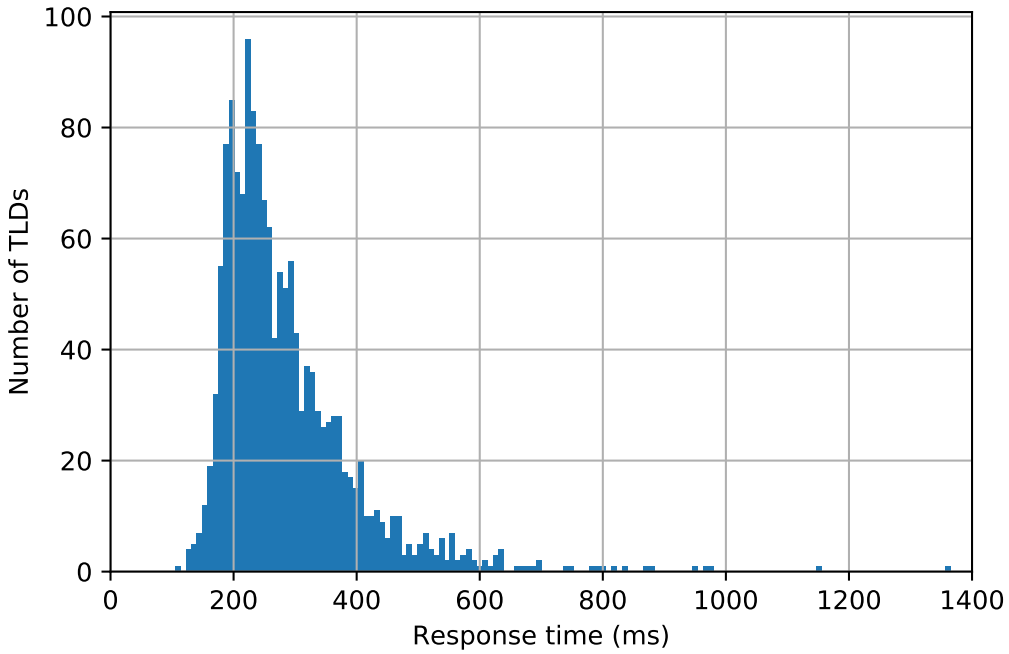

Figure 5.7: Average response times of all TLDs

[1] https://atlas.ripe.net/docs/udm/#rate-limits

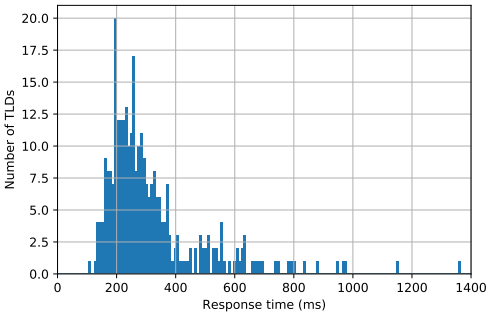

Figure 5.8: Average response times of old TLDs

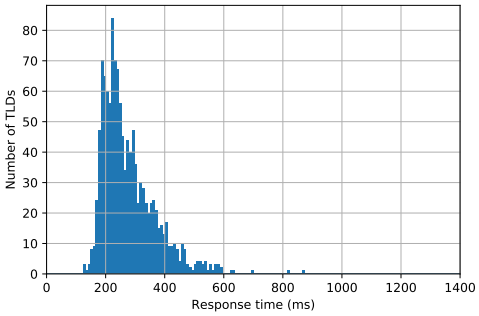

Figure 5.9: Average response times of new TLDs

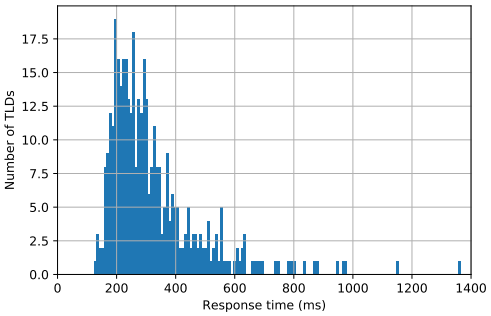

Figure 5.10: Average response times of ccTLDs

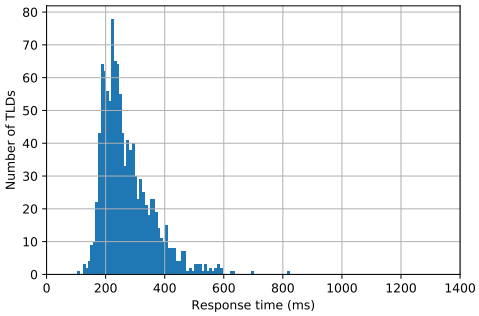

Figure 5.11: Average response times of gTLDs

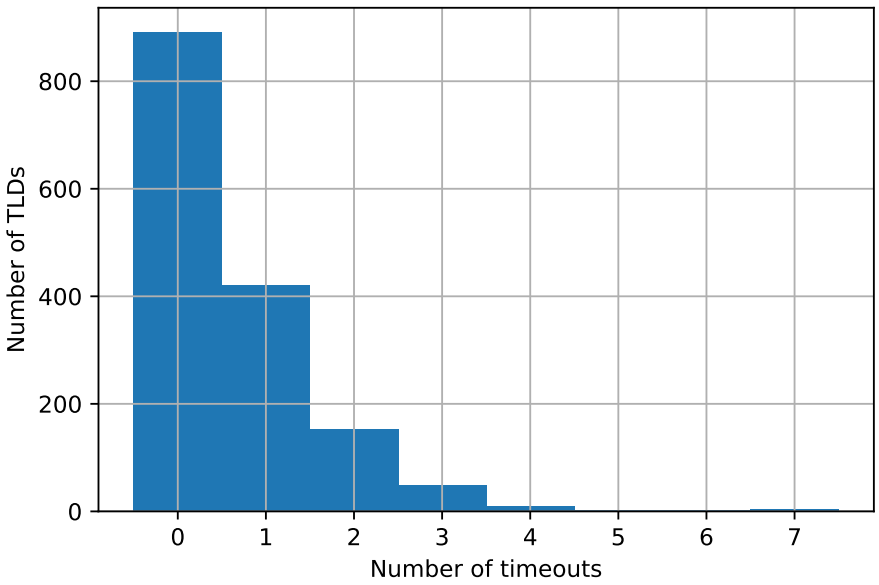

Figure 5.12: Distribution of number of timeouts over number of TLDs.

5.3 Robustness

5.3.1 Network diversity

The histogram in figure 5.13 shows the number of TLDs hosted within each AS that hosts at least one TLD. The vast majority of ASs only contain 1 TLD.

In figure 5.14 the histogram shows the frequency of the number of unique ASs in which a TLD is hosted in. These frequencies range from 1 to 15. The highest frequency occurs at when TLD is hosted in 2 ASs closely followed by 3 after which it decreases.

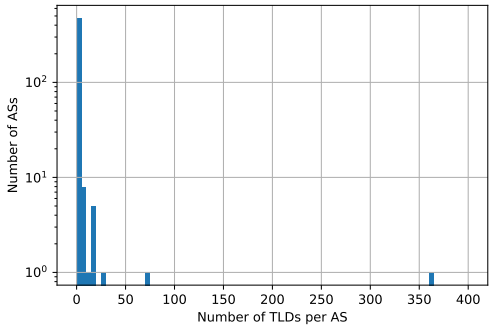

Figure 5.13: TLD spread over ASs

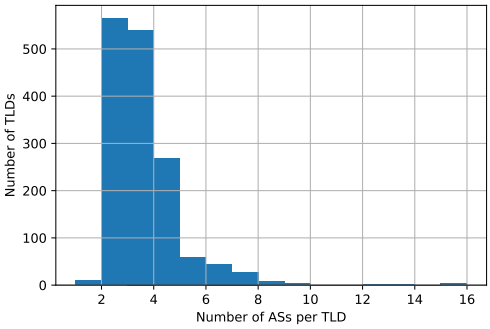

Figure 5.14: AS spread over TLDs

5.3.2 Reachability

Figures 5.15, 5.16, 5.17, 5.18 and 5.19 show the distribution of what protocols are supported by the name servers. The two protocols TCP and UDP are tested. There are 4 possible outcomes: a name server supports both TCP and UDP, a name server supports only TCP, a name server supports only UDP or a name server does not support TCP and UDP.

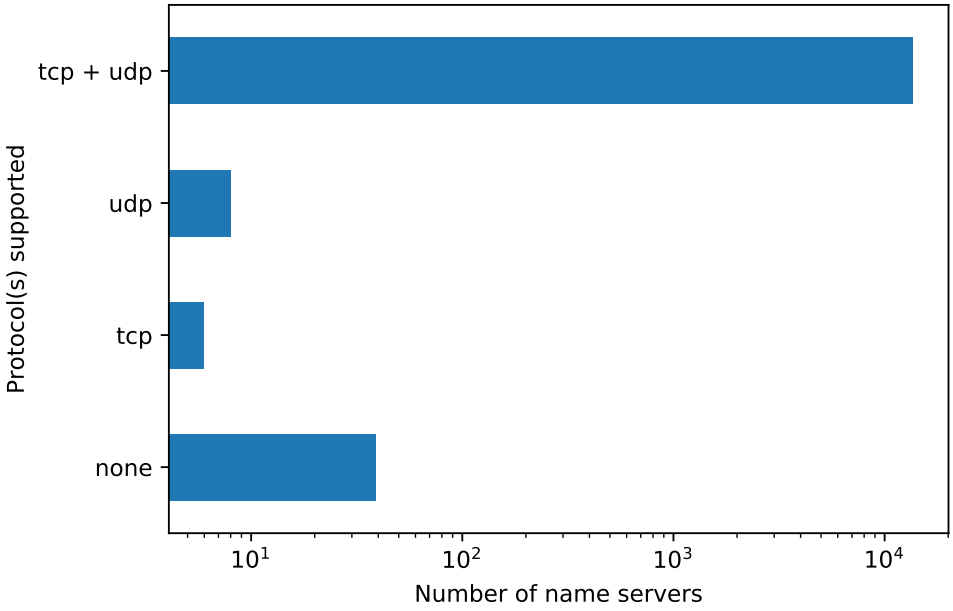

Figure 5.15: TCP and UDP support of all name servers.

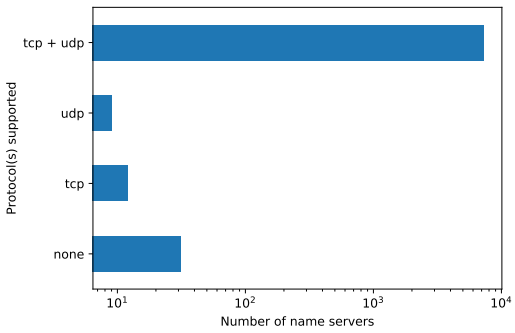

Figure 5.16: TCP and UDP support of IPv4 name servers.

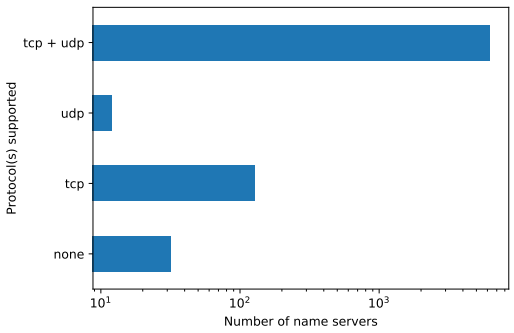

Figure 5.17: TCP and UDP support of IPv6 name server.

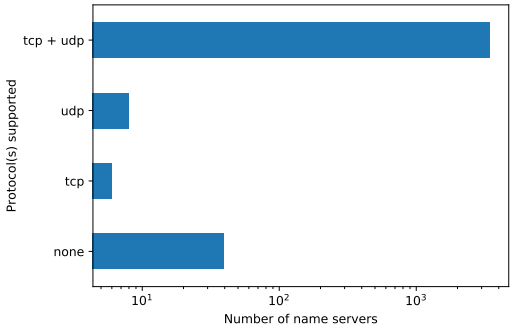

Figure 5.18: TCP and UDP support of old TLD name servers.

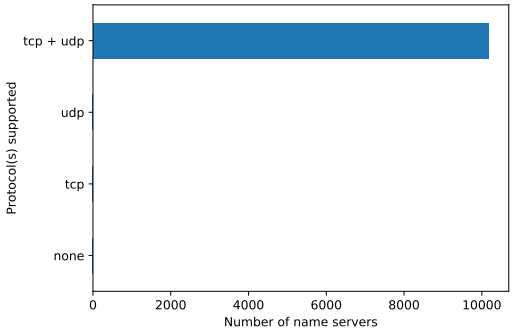

Figure 5.19: TCP and UDP support of new TLD name servers.

5.3.3 Name servers

The histogram in figure 5.20 illustrates the frequency of the number of name servers each TLD has. The most frequent number of name servers for a TLD occurs at 4 name servers.

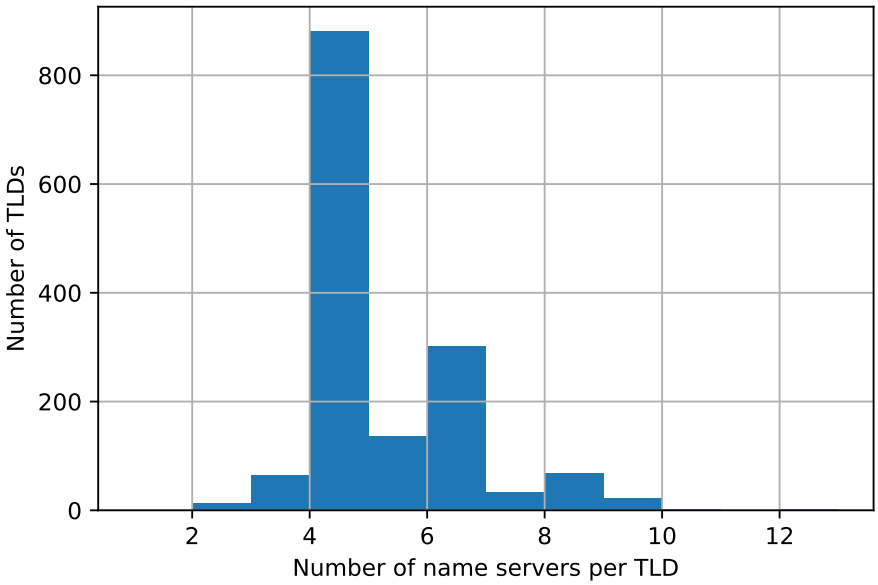

Figure 5.20: Frequency of number of name servers per TLD.

CHAPTER 6

Discussion

6.1 Credibility

Table 2.1 show that according to the current standard RSASHA1 falls under the category of *Must Implement*. The algorithms RSASHA1-NSEC3-SHA1, RSASHA256 and RSASHA512 fall under the category of *Recommended to Implement*[12]. However, due to recent exposures of vulnerabilities in SHA1[13], a new Internet draft is aiming to retire the use of RSASHA1 and moving RSASHA256 to be the new must be implemented[1]. This might explain why RSASHA256 is so prominent in figure 5.1. RSASHA512 is the least popular of three algorithms in use. This could be caused by that RSASHA512 requires more data per packet which might not be optimal for some TLDs. It is important to note this is all implementation advice and is not a requierement.

Figure 5.2 shows that a few TLDs exist that have `DNSKEY` record but no `DS` record. This could be because they do not have DNSSEC fully functional and are still working on it. More TLDs could work on getting DNSSEC functional as well as slightly less than 200 TLDs have no DNSSEC related records at all. Figures 5.5 and 5.6 show that these are all ccTLDs since every gTLD has implemented DNSSEC.

6.2 Performance

Figures 5.7, 5.9, and 5.11 show that all TLDs, new TLDs and gTLDs have the same distribution of average response time per TLD. They all have their maximum around 200 ms and decrease at a similar rate in both directions from that maximum. The performance of gTLDs and new TLDs almost looks the same which is not that surprising considering most new TLDs are gTLDs. Figure 5.8 and 5.10 have a different distribution than the others, because they generally performs worse. Though the distribution gTLDs and new TLDs look very much the same, ccTLDs and old TLDs do not. The distribution of ccTLDs more closely resembles the other whereas the old TLDs are spread out more equally.

Most end users are willing to wait around 2 seconds for simple information retrieval from the web[11]. Getting the the IP address of the second-level domain should definitely not last longer than 1000 ms as some TLDs do since the process of information retrieval from the web encompasses more than just querying the name server of the TLD for the second-level domain. DNS requests need to be send to multiple DNS servers and then the web page needs to be loaded. Although it should be noted that this is one time process because the address gets cached afterwards.

Some leniency can be applied to certain TLDs though, because they are country related like ccTLDs but also gTLDs that are only related to a specific country and or language. Since the use of these TLDs is mostly isolated within specific regions, performance outside these regions is less relevant for these TLDs. Additionally, in order for these TLDs to have a decent response time globally would require a lot of investment in additional infrastructure. These investments in infrastructure cost a lot of money which causes decision making on whether is worthwhile or

not.

A minor problem with the performance is that the average response time per TLD is that of the different probes, but not multiple measurements on each probe. Though this problem is somewhat mitigated by using multiple probes. For a better representation of the response time it might be better to run the same test on each probe multiple times at different times. It was however not possible to do this as it would vastly increase the cost of each measurement since RIPE Atlas requires credits for measurement.

The timeouts in figure 5.12 show that around 900 TLDs have no timeouts which means that around 600 TLDs are having issues in being reachable from some probes within 5 seconds. Although for some TLDs this could be a one time problem because the measurements were only performed once per TLD. It is however apparent that some TLDs do have major problems if 7 out of 75 probes end up timing out.

6.3 Robustness

6.3.1 Name servers

Figure 5.20 shows that every TLD has at least two name servers which is in accordance with the technical requirements for authoritative name servers by IANA[1]. As such not a single TLD is dependent on just a single name server being reachable. There there is the possibility that multiple addresses are being used by the same name server though, but this seems unlikely. It is not certain however if these are the actual number of name servers since it is possible they are anycasted. Overall, the vast majority of TLDs have a decent amount of name servers, but it is meaningless if these name servers are not spread throughout topologically separate networks.

6.3.2 Network diversity

The histograms from figures 5.13 and 5.14 both show positive and negative aspects of the network diversity. One of the more positive aspects is shown in figure 5.13 where the vast majority of ASs have only around one TLD. This means that TLDs are spread out over many unique networks and as such the impact of one of these networks going down does not affect TLDs as a whole. There is one network however that has around 360 TLDs hosted within it. However, if figure 5.14 is considered as well, it shows that most TLDs are spread over 2 to 3 networks. So even if this AS with around 360 TLDs were to fail, it will most likely mean that the TLDs in this network still remain reachable as they are also hosted on other networks. The technical requirements for authoritative name servers[1] states on network diversity that the name servers must be in at least two topological different networks. Almost all TLD name servers follow this requirement except for a small group as shown in figure 5.14. However, it could be that the name servers in this group are anycasted which would make them comply with the requirement.

6.3.3 Reachability

Figure 5.15 shows that almost every name server IP address supports either TCP or UDP and that the vast majority supports both. It also shows that some name server IP addresses do not support TCP and UDP which practically makes a name server useless. Although this does not pose any direct problems because as shown by figure 5.20 every TLD has multiple name servers. Name servers that do not support both TCP and UDP should be fixed though or removed from the `NS` records, because they violate the technical requirements[1] that state that a name server must answer queries over both TCP and UDP. Figures 5.18 and 5.19 shows that this problem originates from old TLD name servers since every new TLD name server supports both TCP and UDP.

When comparing figure 5.16 and 5.17 it shows that the only major difference is that the support of TCP. This means that UDP is supported less over the IPv6 addresses of name servers. This is not really a problem, because almost all DNS queries exceed the maximum size for UDP and as result they mostly use TCP.

[1] https://www.iana.org/help/nameserver-requirements

CHAPTER 7

Conclusion

In conclusion, from the results it is apparent that in general most TLDs perform well in credibility, performance and robustness. This is shown for credibility because the vast majority of TLDs supports DNSSEC, but this does not mean that their delegations are DNSSEC signed. The average response time of most TLDs is less than 400 ms showing that overall performance is good. The robustness of TLDs is illustrated by the network diversity which shows that most ASs only host a single TLD. Also almost every TLD uses at least two ASs and every TLD has at least two name servers. The support of TCP and UDP over IPv4 and IPv6 is present in most TLDs as well.

However, it is apparent that new TLDs perform better in general and the worst cases all occur in its counterpart, old TLDs. The same is the case for gTLDs and ccTLDs, but gTLDs do not outperform ccTLDs as much as new TLDs outperform old TLDs in every property. This means that the new gTLD program has been successful into introducing new TLDs of good quality to the DNS and that some old TLDs are not keeping up with the current state of the art.

The framework presented by this thesis is successful in determining the quality of TLDs and identifying TLDs that are performing poorly. Cases of poorly performing TLDs are found in credibility where some TLDs do not support DNSSEC, in performance where some TLDs had average response time of over 1000 ms and in robustness where some TLDs did not support the protocols TCP or UDP.

7.1 Future work

Although the framework is successful in determine the quality of TLDs, there is still room for improvement by increasing the accuracy of some of the results like performance and extending the number of results by including more properties. The accuracy of the performance results could be improved by performing measurements on each probe various times and at different times of day. As well as using more probes with RIPE Atlas and/or other tools such as Periscope[5] and combining them for more accurate results. Another improvement is extending the data gathered. Properties such as software diversity can be measured by fingerprinting name servers. This could provide more insight into robustness, because in the event that one name server software has a vulnerability it is important that not all name servers of TLD become comprised due to this one vulnerability. Additionally, privacy can be measured by looking at the routing tables of DNS requests to see to which networks your requests is available to. Also, the WHOIS data of domains within each TLD zone could be analysed to determine the availability of personal information by each TLD. It could also be interesting to tag each TLD for which country or language they are used to see if this has any influence on global performance. The credibility results could be extended by including the adoption rate of signed delegation within each TLD zone. Finally the method proposed for anycast detection could performed to detect which TLD name server use anycast.

Bibliography

[1] R. Arends, J. Schlyter, and M. Larson. DNS Security (DNSSEC) DNSKEY algorithm IANA registry updates. Internet-Draft draft-arends-dnsop-dnssec-algorithm-update-00, Internet Engineering Task Force, March 2017. Work in Progress.

[2] N. Brownlee, K. Claffy, and E. Nemeth. DNS measurements at a root server. In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 3, pages 1672–1676. IEEE, 2001.

[3] N. Brownlee, K. Claffy, and E. Nemeth. DNS Root/gTLD performance measurements. *USENIX LISA, San Diego, CA*, 2001.

[4] D. Eastlake. Domain name system security extensions. RFC 2535, RFC Editor, March 1999. `http://www.rfc-editor.org/rfc/rfc2535.txt`.

[5] V. Giotsas, A. Dhamdhere, and K. Claffy. Periscope: Unifying looking glass querying. In *International Conference on Passive and Active Network Measurement*, pages 177–189. Springer, 2016.

[6] J. Haas and J. Mitchell. Reservation of last autonomous system (AS) numbers. BCP 6, RFC Editor, July 2014. `https://www.rfc-editor.org/rfc/rfc7300.txt`.

[7] T. Hardie. Distributing authoritative name servers via shared unicast addresses. RFC 3258, RFC Editor, April 2002. `http://www.rfc-editor.org/rfc/rfc3258.txt`.

[8] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an autonomous system (AS). BCP 6, RFC Editor, March 1996. `https://www.rfc-editor.org/rfc/rfc1930.txt`.

[9] ICANN. gTLD applicant guidebook. Technical Report 1, ICANN, 9 2011. `https://archive.icann.org/en/topics/new-gtlds/intro-clean-19sep11-en.pdf`.

[10] J. Liang, J. Jiang, H. Duan, K. Li, and J. Wu. Measuring query latency of top level DNS servers. In *International Conference on Passive and Active Network Measurement*, pages 145–154. Springer, 2013.

[11] F. Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004.

[12] S. Rose. Applicability statement: DNS security (DNSSEC) DNSKEY algorithm implementation status. RFC 6944, RFC Editor, April 2013. `https://www.rfc-editor.org/rfc/rfc6944.txt`.

[13] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov. The first collision for full SHA-1. *URL: https://shattered. it/static/shattered. pdf*, 2017.

[14] D. Treise, K. Walsh-Childers, M. Weigold, and M. Friedman. Cultivating the science internet audience: Impact of brand and domain on source credibility for science information. *Science Communication*, 24(3):309–332, 2003.