UNIVERSITY OF AMSTERDAM

Master in System and Network Engineering

Analysis of DNS Resolver Performance Measurements

Author:
Hamza Boulakhrif
Hamza.Boulakhrif@os3.nl

Supervisors:
Yuri Schaeffer
Yuri@nlnetlabs.nl
Willem Toorop
Willem@nlnetlabs.nl

July 13, 2015

Abstract

The Domain Name System (DNS) is an essential building block of the Internet. Applications depend on it in order to operate properly. Therefore DNS resolvers are required to perform well, and do whatever they can to provide an answer to a query.

In this paper a methodology is devised to measure the performance of the Unbound, BIND, and PowerDNS resolvers. Measurements of these resolvers is required to be objective and not biased. Analysis is conducted on these performance measurements, where the implementations are compared to each other. Corner cases have also been identified and analysed.

Acknowledgement

I would like to thank NLnet Labs for providing this interesting Research Project. Also the support they provided throughout this project is really appreciated. Especially Willem Toorop and Yuri Schaeffer who answered all of my questions and guided me through this project. Thanks go also to Wouter Wijngaards who answered my questions related to DNS resolvers.

The performed measurements during this project were carried out from the SNE lab, at the University of Amsterdam. I would also like to thank Harm Dermois for providing his server in the lab to perform these measurements.

Special thanks also go to the SNE staff for their support and guidance. This does not only concern this project but the entire study.

Hamza Boulakhrif

Contents

Chapter 1

Introduction

The Domain Name System (DNS) is an Internet building block and service for the translation between names and IP addresses and nowadays more than that. DNS is a hierarchical distributed naming tree where domain name owners are responsible for their part of this tree. This distributed naming system is used by DNS resolvers to provide resolving services to users and applications. These services are assumed to provide users with a response as quick as possible.

The most important and basically only components in the DNS architecture are DNS authoritative servers and DNS resolvers. There has been research on measuring the performance of DNS authoritative server implementations where other methodologies have been devised to perform such measurements, like the DISTEL methodology [6]. DNS resolvers on the other hand have not been researched as extensive as DNS authoritatives when it comes to performance. This is due to the fact that DNS resolvers are inherently difficult to measure because of their recursive behaviour.

This research focuses on the analysis of performance measurements of the well known and widely used DNS resolvers Unbound, BIND, and PowerDNS. Before measurement and analysis has been performed, a methodology has been devised for performance measurements of these different implementations. This methodology provides a way to measure different resolver implementations in an objective way. It also provides a way to easily extract measurement data.

This report is structured as follows: In chapter 2 an overview of DNS and DNS resolvers is given. Then chapter 3 describes the methodology devised for the measurements of DNS resolvers. Chapter 4 presents the analysis of the measurement results. The conclusion and future work are described in chapter 5 and 6 respectively.

1.1 Related Work

A number of papers are available on DNS server measurements and methodologies of such measurements. This includes both DNS authoritatatives and resolvers.

Existing methodologies can be leveraged and extended in order to fit this research. Methodologies used for measuring DNS authoritatives measure performance in terms of amount of queries per second [6]. Although DNS resolvers have been measured in the same way as

DNS authoritatives [12], there are also methodologies that measure resolvers in terms of time per query [5].

Research that has been performed on analysis of DNS authoritative server measurements, focus on various areas. Papers are available of performance measurements on specific DNS authoritative servers, related to a role or company [10]. Performance measurements have also been conducted for specific DNS authoritative implementations [3] [7].

When it comes to DNS resolvers, research is widely available on measuring specific resolvers of certain companies. Like the research that has been done on the performance of Google DNS, OpenDNS, and DNS resolvers of local ISPs [1]. However, research on performance measurements of DNS resolver implementations is present to a much lesser extent. Only one paper was found that performs measurements and analysis (to a certain degree) on DNS resolver implementations [12]. This paper uses a methodology that was initially devised for the use with DNS authoritative servers.

1.2 Research Questions

The research is formed around the main question shown below.

What is the performance of various DNS resolver implementations?

- Can a methodology be devised to measure the performance of DNS resolver implementations objectively?
- Can corner cases be identified by comparing DNS resolver measurement results?

1.3 Approach and Methods

The approach for this project consists of a number of steps to allow the research to be performed in a structured way. Firstly, existing papers were sought that are about DNS performance measurements and performance analysis of DNS. This should give insight in what has already been done but also to build upon.

Secondly, a methodology is devised to perform the measurements on DNS resolver implementations. Use is made of existing methodologies in order to achieve the goal of this project and extend these methodologies wherever needed.

The environment which is used to performed the measurements is then set up. This includes physical components, software, tools, programming, and more. After these are set up, the measurements are ready to be performed.

Lastly, analysis is performed on the measurement results. Uncovering corner cases is performed after the analysis.

Chapter 2

Background

DNS is a distributed system which consists of key value pairs for the translation of names to addresses. This is organized in a tree structure, which is used for the lookup process.

DNS is one of the most important services on the Internet and is used daily by users all over the world. DNS supports a lot of applications including web browsing, e-mail, chatting, and a lot more. The name system therefore does not only need to be operational all the time but also needs to provide resolution as quick as possible in order to provide a pleasant use of the applications that rely on it.

The DNS service basically consists of two essential components. These are the DNS authoritative servers and the DNS resolvers. While the authoritative servers contain the DNS records, the resolvers perform the lookup of these records.

Beside to the lookup process, DNS resolvers also provide caching in order to make the resolution faster. Validation is done by resolvers too, which is necessary for DNSSEC, the security extension of DNS.

The resolution of a domain name takes a number of steps before an answer can be provided to the querier. A query is normally sent by a user or an application to the local resolver. The resolver then tries to resolve the domain name by contacting a number of authoritative servers. For many of the queries a single authoritative server does not provide the final answer. Authoritative servers refer resolvers to other authoritative servers that know more about the requested query. This goes on until the resolver contacts an authoritative server that is able to answer the query. In figure 2.1[1] an example is depicted of the process of resolution from the DNS resolver's point of view.

It all starts when a user tries to navigate to a website, which is in this case "www.mybank.com". It is the first time this user visits this website and therefore does not have the IP address in its cache. This means the web browser first requires resolution of the domain name before it can download the website.

The browser sends a query to the (local) DNS resolver in order to perform resolution. The DNS resolver receives this query and should do its best to resolve the query recursively by contacting the right authoritative servers.

In order for a DNS resolver to perform the lookup process it needs to know what DNS

[1] http://www.technicalinfo.net/papers/Pharming.html

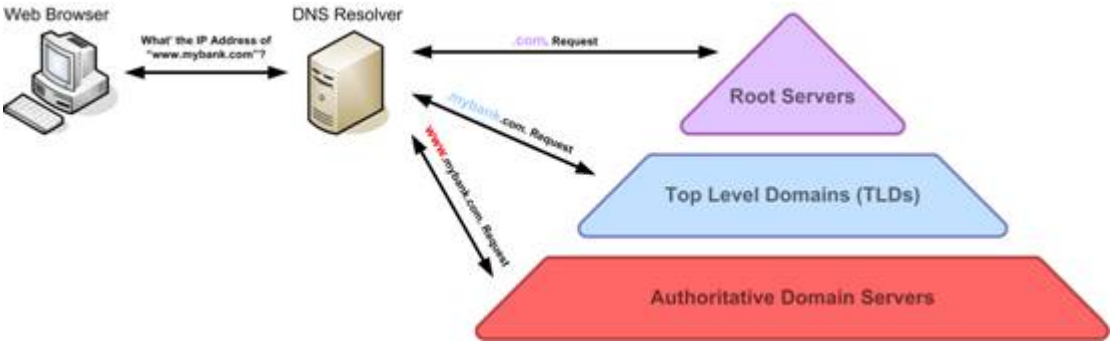

Figure 2.1: DNS resolution process

servers to contact. Every DNS resolver (that is connected to the Internet) is populated with a root file which contains all DNS root servers for bootstrapping purposes. These root servers at the top of the DNS distributed tree are the starting point for resolution. When certain authoritative servers have been cached, the root servers can be skipped in the resolution process.

To resolve "www.mybank.com", a number of authoritative servers have to be queried. These queries will result in increasingly more specific referrals to the authoritative server for the domain name. This goes on until an answer is found to the query. The resolver first contacts the root servers which are denoted as a dot (.), which are at the top of the DNS tree. The root server does not know the answer but does have a more specific referral to get to the answer. A referral is sent back to the resolver that point to the ".com" top-level domain authoritative servers. Although not common, It is possible for authoritative servers to resolve the query recursively instead of sending a referral to the resolver.

Next, the resolver tries to contact the ".com" authoritative servers to resolve "www.mybank.com". These servers do not have the answer but do have a more specific referral to the authoritative server of the domain. A reply is sent to the resolver again with a referral with information of where to contact the name server of the "mybank.com" domain.

Lastly, the resolver contacts the "mybank.com" servers to resolve "www.mybank.com". These servers have a resource record present containing an IP address for "www.mybank.com". A reply is sent to the resolver. Because this reply contains an actual answer, this is sent and presented to the application that made the originating query. The user is now able to download the website by contacting the web server via its IP address.

Note that during the whole process of resolution the DNS resolver caches answers to provide users and applications with a faster answer.

A number implementations exist that perform the tasks of a DNS resolver. Amongst other aspects it is important that these resolvers perform well and try hard to provide an answer

to queries. Well known, widely used and Open Source resolvers are Unbound of NLnet Labs, BIND of the Internet Systems Consortium and PowerDNS of PowerDNS.COM BV.

Chapter 3

Methodology

In order to measure DNS resolvers in a objective manner, a methodology is required. As described in chapter 2, DNS Resolvers perform lookups of queries in a recursive manner. This means queries sent to these resolvers need to be resolved on behalf of the clients first, before an answer is given to the user or application that requested the query in the first place. The recursive lookup that resolvers perform makes measuring them an inherently difficult task.

This chapter describes the methodology used to measure the DNS resolver implementations. This includes looking at existing methodologies but also describing the devised methodology to perform the measurements. Also part of this chapter are the tools and dataset used during this research and the methodology for the analysis.

3.1 Existing Methodologies

Other papers that focus on measuring DNS resolvers, have devised methodologies to perform such measurements. These papers primarily focus on either benchmarking, which is all about the number of queries, or latency, which is about the time it takes to resolve a query. These methodologies are looked at to help devise a methodology that can be used for the measurements.

The benchmarking methodology is a methodology that is more often seen when it comes to measuring DNS authoritative servers. The DISTEL methodology is such a methodology that is based on benchmarking. DISTEL is often used to measure various authoritative server implementations and compare their results with each other. This methodology is also observed to be used with DNS resolvers [12]. What basically happens is that a big number of queries is sent to the resolver in order to find out how much it can handle per time unit. Measurements is then performed based on queries sent versus responses received. This methodology therefore measures in terms of queries (sent/received) per second.

This methodology appears to work well when measuring DNS authoritative servers because they (usually) do not operate in a recursive manner. When queries are sent to an authoritative server it can provide an answer from the data is possesses. This is different as opposed to DNS resolvers which do operate recursively and might not have the answer to

queries. Resolver therefore have to contact a number of authoritative servers for an answer to the query.

When benchmarking resolvers one has to be careful that external factors do not to accidentally influence the measurements. This could happen due to other elements such as the networks or other DNS servers. Especially when DNS resolvers are measured on the Internet, this can become an issue. This is due to limitations and bottlenecks on certain points on the Internet, which are hard to identify and take in to account in the measurements.

Other papers describe a methodology that conduct performance measurements on DNS resolvers in terms of time per query. This methodology is all about tracking the time queries arrive at resolvers and when replies are received. The delta of both is the resolution time of the query i.e. the latency. This measurements is a more granular methodology and looks at each and every query performed.

This methodology, which is based on latency per query, can also be influenced by surrounding elements, as is also the case with the benchmarking methodology. With benchmarking however, this influence is hard to limit and control. Using the latency methodology, influence of external factors on measurements is more limited because it does not stress these elements.

3.2 Requirements Methodology

Before a concrete methodology can be devised to perform measurements on DNS resolver implementations, one has to identify what the requirements are for such a methodology. These requirements are mainly based on the research questions described in chapter 1.

Because measurements are performed using different DNS resolver implementations, a requirement for a methodology is the possibility to use it with various DNS resolver implementations. The methodology also needs to provide a way to compare measured resolvers without bias.

As described in chapter 1 Unbound, BIND, and PowerDNS are the DNS resolvers used for the measurements. It is important that these resolvers are measured objectively and no biased measurements are present. The measurements would also be more useful if these are based on reality, to prevent a distorted picture of DNS resolvers. The Internet is a perfect way to accomplish this. The Internet also allows a wide range of diverse DNS resource records that can be leveraged for the measurements. In order to leverage the DNS resource records on the Internet, one needs representative DNS queries over synthetic data.

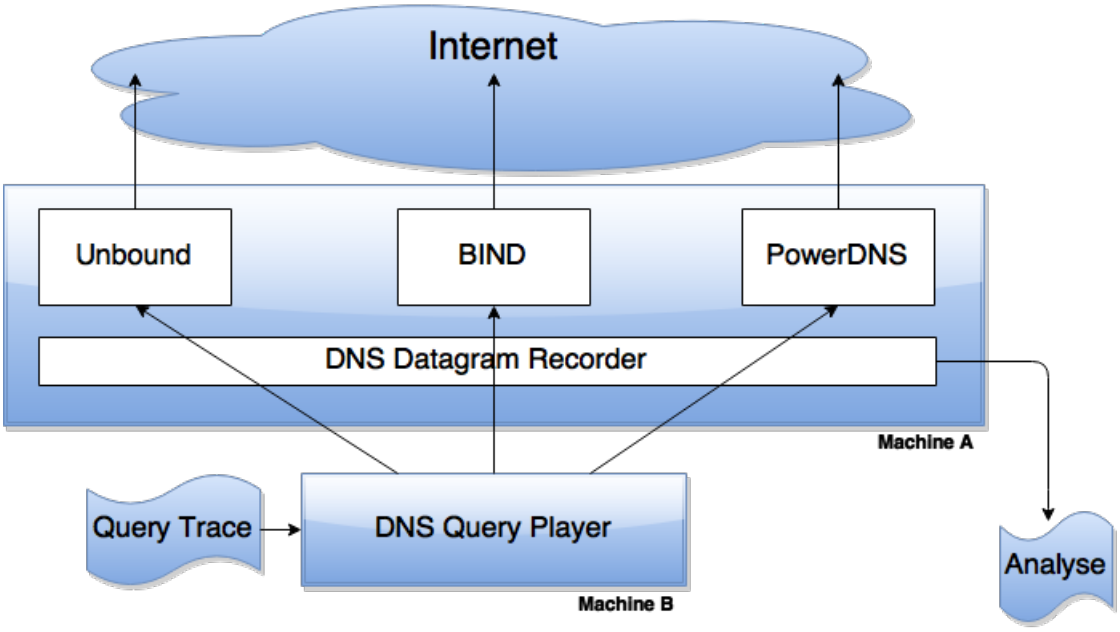

Figure 3.1: Measurement Methodology

A solution to these requirements is given in section 3.3 where the devised methodology is described.

3.3 Measurement Methodology

In this section the methodology is described that is devised for the measurements of various DNS resolver implementations. This methodology bases its measurements in terms of latency and complies with the requirements identified in section 3.2.

The requirements form the basis for choosing the methodology based on latency per query. When performing measurements on resolvers that leverage the Internet, the Internet itself might influence the measurements too much. Especially when measuring more than one resolver the problem gets worse because the resolvers might influence each other. Figure 3.1 shows a diagram of the devised methodology.

Figure 3.1 shows a number of components. Although two physical machines are drawn in the figure, the DNS Query Player can be placed on the same machine as the DNS Datagram Recorder. Due to limitations of the tools used in section 3.5, two machines are used.

The DNS Query Player is the component that sends the queries towards the resolvers. This

is performed in such a way that each query goes to all resolvers at the same time. The query trace is the one that provides a lot of queries, which the DNS Query Player sends to the resolvers.

The DNS Datagram Recorder on the other machine records all passing by DNS traffic. This is necessary to figure out how long each and every query took to resolve by each DNS resolver. After the query trace is completely played and the DNS Datagram Recorder has stored all data, the data is extracted from the DNS Datagram Recorder and used for analysis.

The resolvers that receive the queries from the DNS Query Player, send the queries to authoritative servers all over the Internet. In order to make the measurements between resolvers comparable, it is necessary that the queries are sent to the resolvers at the same time. The necessity to send the queries at the same time comes from the changing nature of the Internet. By this way the influence of changes on the Internet is considered and kept to a minimum.

3.4 Dataset Methodology

The dataset used i.e. query trace is an important factor in performing the measurements. Although a dataset does not represent the whole DNS system it does give a good representation of the name system. It should also be noted that not only the Internet is changing but also DNS authoritative servers can change. This means that the measurement results performed with this dataset, are likely to change over time. Despite this, for comparison purposes this does not really have an impact. Even though the "numbers" might change, the differences between resolvers stay the same.

The dataset used needs to comply with the requirements described in section 3.2. This means that the dataset needs to contain diverse queries but also needs to be as realistic as possible. The dataset used for the measurements comes from Nominum [9]. They provide a query trace of 10 million queries of which almost 374 thousand are used. The amount of queries used is limited by the amount of time a measurement takes. Using the whole dataset would mean measurements of more than a day.

3.5 Measurement Tools

The tools used for the measurements and extraction of data are described in this section. Existing and well known tools are preferred over programming new tools. Not only because one does not want to reinvent the wheel but these tools also have proven to work well. Only in the case where no existing tools are available, tools are created.

For the DNS Query Player shown in figure 3.1, Tcpreplay is used. Tcpreplay is a well known suite of Open Source tools for editing pcap files in libpcap format and replaying these on network devices [2]. It allows one to regulate the speed at which these queries are played towards the resolver in order to prevent overloading components. It also allows sending queries to different destinations simultaneously[1]. The query trace file that is injected into the DNS Query Player is in pcap format. This makes that Tcpreplay is the ideal tool to use for this purpose.

The tool that is used for the DNS Datagram Recorder is tcpdump, which is also a well know and Open Source tool. Tcpdump is a command-line packet capturer and analyzer [8]. It is used to capture both the query and response in pcap format for later analysis. The pcap file format is ideal for this purpose because all data one could need is available and the necessary means to extract data from pcap files are also available.

After measurements are performed one needs to use the pcap with results from the DNS Datagram Recorder for data extraction and visualisation. Unfortunately no ready made tools exist that are able to perform these tasks, which is why a program is created to do so.

For the extraction and visualisation of data from pcap files, the Python programming language is used. Python is used because it is an straight forward language with a lot of libraries. It is also a language where the researcher is familiar with.

For the extraction of (DNS) data, the DPKT library is used, which is a packet creation and parsing library for TCP/IP protocols in Python [11]. This library had some limitations for parsing certain fields in the DNS protocol. Especially DNSSEC is not implemented (yet). A manual solution is programmed instead.

The Matplotlib library is used for visualisation of data for analysis [4]. Besides these two mainly used libraries the Numpy, Socket and Sys libraries are also used to perform array manipulation, formatting extracted data and argument handling from the command line respectively. The created Python program can be found in Appendix A.

3.6 Analysis Methodology

Before analysis is performed on the measurement results, a methodology is conceived. This is in order to figure out how analysis is going to be performed but also to exclude certain forms of analysis. It is absolutely not the intention to fix the way data is analysed but rather for guidance.

[1] There is no such thing as sending traffic simultaneously over the same network. The differences in time of sending identical queries to different resolvers are extremely low, and each pair of query and response are measured separately

In order for results of measurements to have (more) meaning, they are compared with each other. But there are limitations to comparing certain measurement results. One should not compare measurements that are conducted on different points in time because it could lead to wrong conclusions. The difference between results of these measurements may be caused by changes on the Internet or DNS authoritative servers.

The methodology of choice is a comparison between measurements of different DNS resolvers that have been performed simultaneously. This limits the influence of other factors such as the Internet but still allows one to perform analysis. This methodology can also be used in conjunction with a breakdown of the measured results.

Chapter 4

Analysis

In this chapter the results of the measurements are analysed. These results are achieved by using the methodology described in chapter 3. This chapter is divided in measurements performed with DNS and DNSSEC. It also contains corner cases found during these measurements.

An important factor to mention is the configuration of the DNS resolvers. A resolver has a large number of variables that can be tweaked in order to manipulate the performance of a resolver. The results of the measurements that are shown in this chapter are performed with a default configuration of these resolvers. This decision fits the time frame of this project.

Unbound, BIND, and PowerDNS are the resolvers of choice. During the execution of the measurements, the latest stable versions are used of these implementations. PowerDNS increased in version at the time of writing, see table 4.1 for the versions used.

Implementation	**Version**
Unbound	1.5.3
BIND	9.10.2
PowerDNS	3.7.2 (current: 3.7.3)

Table 4.1: Version number per DNS resolver implementation

Another important note is that IPv4 and IPv6 are enabled simultaneously on the resolvers to allow resolvers to make full use of the DNS system.

For the measurements, equipment and facilities are used from the System and Network Engineering lab. This is also the location from where the measurements have been performed. The exact location of measurements is Science Park Amsterdam, The Netherlands as shown in figure 4.1.

4.1 DNS

This section shows the measurements performed for DNS without the security extension (DNSSEC). The measurements performed on the three resolvers are shown below in figure 4.2.

Figure 4.1: Location of measurements

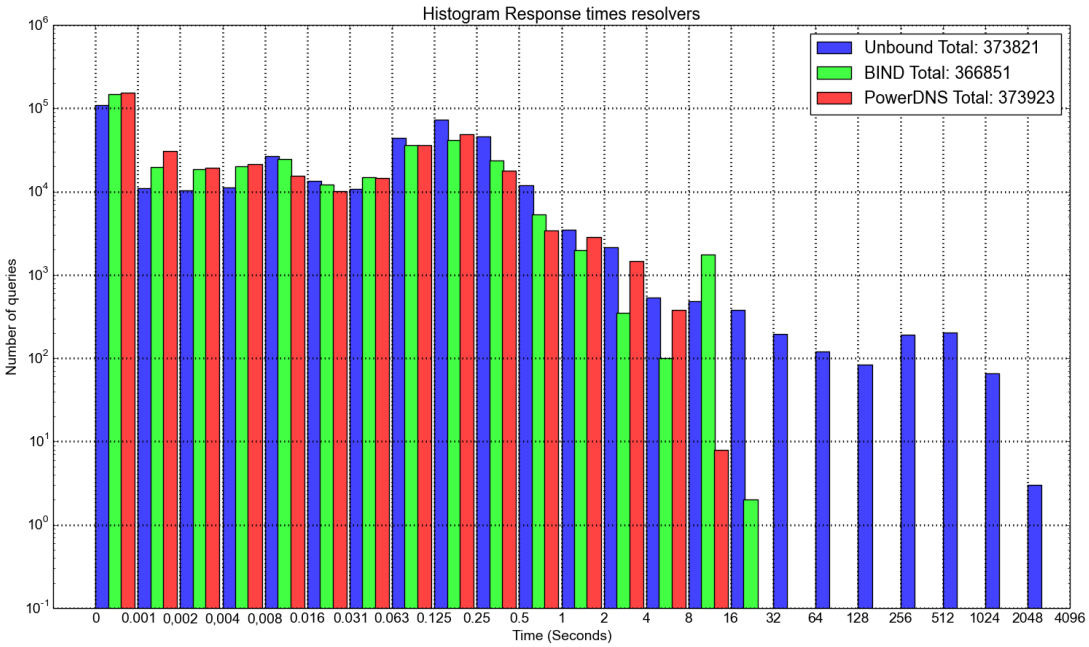

Figure 4.2: DNS resolver measurements

The x-axis, which is an exponential scale shows the latency of queries for each DNS resolver. The y-axis, which is a logarithmic scale shows the amount of queries that are in the time frames shown on the x-axis. The blue, green and red bars represent Unbound, BIND, and PowerDNS respectively. By comparing these results one can understand the differences between the resolvers better.

What can be seen from the figure is the performance of each resolver. It shows how many queries a resolver is able to answer per certain time frame. Between 0 and 1 ms, one can see that PowerDNS followed by BIND answers more queries than Unbound. This time range is likely to be the cache of the resolvers, which means that certain resolvers cache more data than others. Apparently there might be certain resource records that one resolver finds important to cache while the other does not.

The next group of bars that are shown in the plot in figure 4.2 are the ones between 1 ms and 16 ms. Based on their time range, these are likely to come from authoritative DNS servers that are close by to the resolvers. These are likely to be the datacenters that are in close proximity to the measurement site but also countries that are close to the Netherlands such as Belgium. PowerDNS is also in these cases the resolver with the highest amount of responses. BIND again is not too far behind PowerDNS. Unbound on the other hand resolves less than the others and appears to resolve more between 8 ms and 8 seconds. Unbound tends to resolve queries more often at DNS authoritative servers that are more distant. This can be clearly seen between the time frame 63 ms and 1 second. These latencies are most likely to come from inter continental DNS authoritative servers or far away countries. The bars that come after 1 second are likely to come from domain names that are difficult to resolve due to all kinds of reasons, which might be related to DNS or the network DNS uses.

Interesting is also the difference of how long latencies could at most take. Unbound, as can be seen from the plot, takes in a couple of cases between 2048 and 4096 seconds. Converted to minutes this is between 34 and 68 minutes. PowerDNS and BIND take no longer than 16 and 32 seconds respectively. These differences are discussed in detail in the following sections, where each resolver is analysed more in depth.

4.1.1 Unbound

The Unbound DNS resolver data is extracted from the plot shown in figure 4.2 and the same data is shown in figure 4.3 as a stacked histogram below. A division is made in the data based on reply codes to make matters more visible. The only reply codes that are present in the results are no error, nxdomain, and servfail. These were the only ones present in the measurements.

The no error and nxdomain reply codes are both authoritative answers which means that

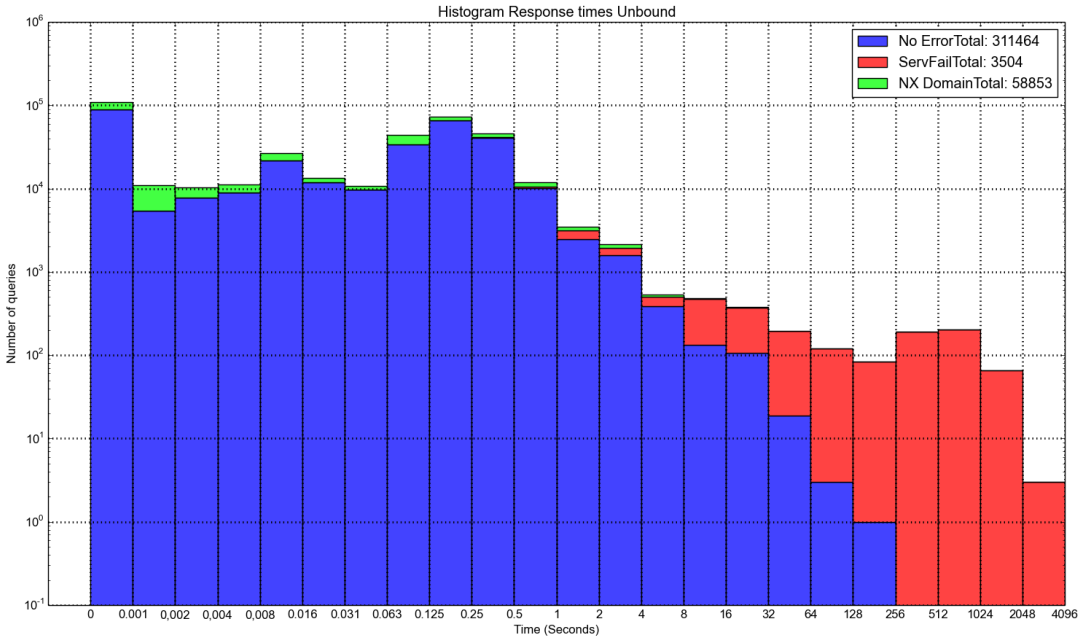

Figure 4.3: Unbound Latency Measurements

the response comes from an authoritative server that is in charge of that domain. Nxdomain is an answer where the requested domain does not exist. With no error everything is basically fine and an answer is returned. Servfails on the other hand are not authoritative answers because no DNS server could answer the query.

The servfails at the right side of the histogram stand out because of their concentration in that area and the logarithmic scale used on the y-axis. When compared to the nxdomains, the servfails are far less in number.

What can be retrieved from the histogram is that Unbound is a resolver that has no timer or at least no short timer for resolving queries. This means that Unbound tries to resolve queries repeatedly until an authoritative answer is acquired. It is assumed that it does this until the chance to resolve these domain names gets very low and then replies with a servfail. This is likely the reason why the servfails are concentrated at the end of the histogram. The fact that Unbound has less servfails compared to other resolvers, supports this analysis. This also tells something about this approach of retries that appears to be successful because Unbound still manages to resolve queries between 128 and 256 seconds. However, the question does remain whether it is necessary to keep trying for over 256/512

seconds. Measurements did not show that queries were answered successfully after this amount of time.

When analysing the histogram, one also observes that the total amount of answers does not equal to the amount of queries in the dataset. Unbound appears to not respond to 75 queries on average. The measurement shown in the histogram did not reply to 102 queries.

Another observation is the peak between 63 and 500 ms. This does not only mean the higher amount of inter continental or more distant resolution of queries but also the different selection of authoritative servers compared to the other resolvers. Resolvers may have different thoughts of what authoritative servers to contact based on performance, security or some other variables.

4.1.2 BIND

The stacked histogram below in figure 4.4 shows the BIND resolver latency data for the measurements performed.

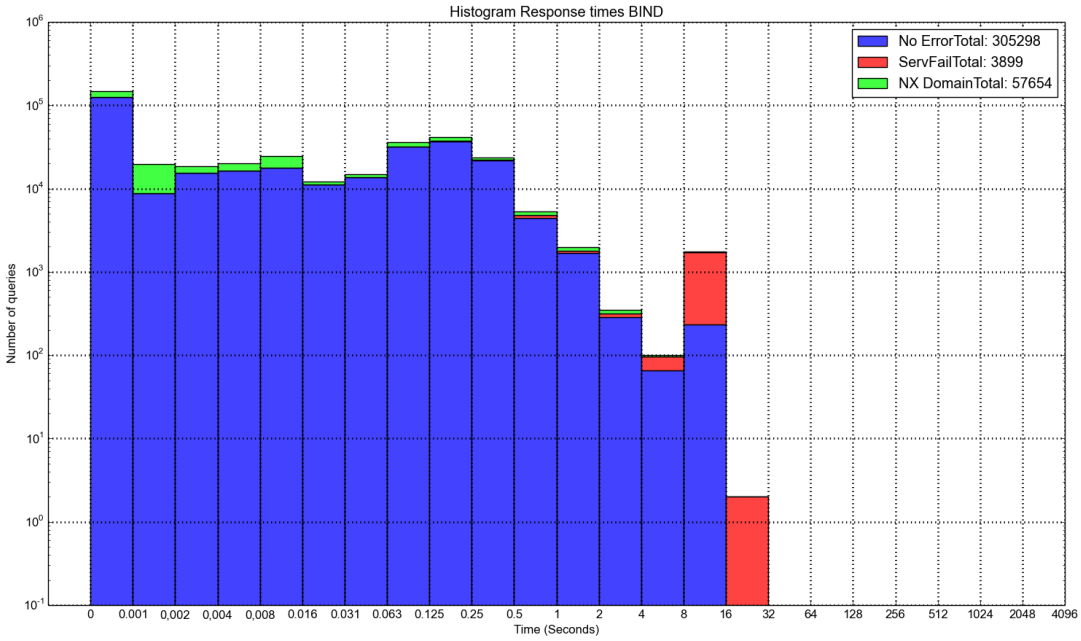

Figure 4.4: BIND Latency Measurements

The measurement results of BIND look much like the ones of PowerDNS and differ from Unbound. What stands out is the right part of the histogram, and the total number of answered queries. BIND is very likely to have a retry timer where each retry has a maximum time to wait for a response. It appears that the maximum a query could take is around 8 - 16 seconds. Some of these take slightly more than 16 seconds which is why the red bar is shown in this time frame. This is a much stricter limit than Unbound poses.

Also interesting about BIND is the fact that it does not always answer the the client. On average BIND does not reply to around 7000 queries, while other resolvers do not show this degree of lost queries. It is not clear why this happens but a resolver normally always replies to the user or application no matter whether it was able to resolve or not. In figure 4.4, BIND did not reply to 7072 of the cases which is easily derived by subtracting the total dataset (373923) from the total amount of answers (366851) found in the histograms.

4.1.3 PowerDNS

PowerDNS, which is the third DNS resolver for the measurements in this research. In figure 4.5 the results of these measurements are shown.

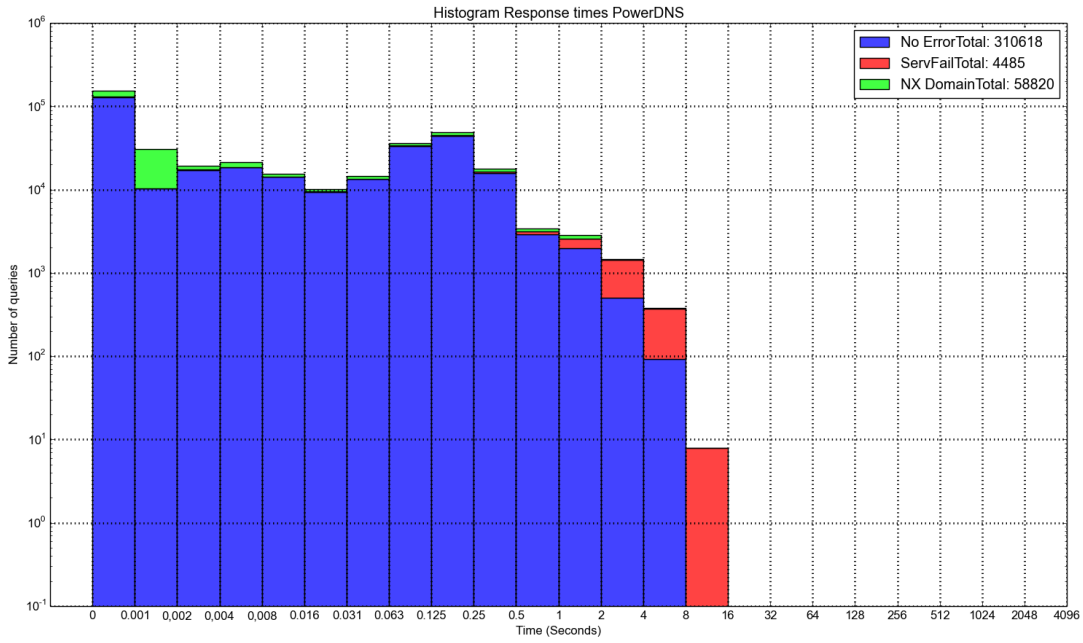

Figure 4.5: PowerDNS Latency Measurements

The measurements that have been performed for Unbound and BIND have also been performed for PowerDNS. The shape of the histogram shown in figure 4.5 looks much like the one for BIND. It is Unbound that differs more from the two. However, PowerDNS is in a lot of the cases able to resolve more in the lower time ranges which can be clearly seen from figure 4.2. This is likely to come from the difference between resolvers in caching behaviour and authoritative server selection.

PowerDNS also applies some sort of timer (like BIND) for recursive queries. Nevertheless PowerDNS appears to have a shorter timer than BIND. The bars in the histogram show that PowerDNS takes at most 16 seconds before a servfail is returned while the measurements of BIND show a maximum of 32 seconds. What PowerDNS does not have is the unlimited timer like the one Unbound uses. What stands out is the number of servfails PowerDNS returns. PowerDNS has a higher number of servfails compared to Unbound with a difference of almost 1000. This is likely to come from the difference between the resolvers in timers where PowerDNS has a greater tendency of returning servfails. Another important reason might be the difference in response to certain queries. For the same query one resolver might give a servfail, while the other resolvers give back a no error without the timers being involved. This is described more in detail in section 4.3.

Interesting to note is the fact that PowerDNS does not lose a single query that is sent to it from the client side. It replies to every query no matter what the results is. This is observed in all measurements performed for PowerDNS.

4.2 DNSSEC

While section 4.1 describes the measurements performed for DNS, this section describes the measurements performed with DNSSEC enabled. As is the case for the DNS measurements, IPv4 and IPv6 are enabled too for DNSSEC on the DNS resolvers to resolve queries towards the Internet.

The three resolvers measured in this paper, do not all support DNSSEC. PowerDNS does not yet have support for DNSSEC while Unbound and BIND do support this. Even though PowerDNS does not support DNSSEC it is displayed in this section for completeness and comparison.

The same dataset is used to perform these measurements as in the case of (plain) DNS. In order to perform these measurements, DNSSEC is enabled on the resolvers (that support it) and the query trace is modified by enabling the AD flag for each DNS query. This flag tells the resolver to perform DNSSEC. It is only able to if the whole chain of authoritative servers for each individual query supports DNSSEC. The amount of queries that resulted in a valid DNSSEC response were on average 4.4 percent for this dataset. This might differ

from different datasets used for measurements but also the Internet as a whole.

In figure 4.6 a histogram is shown of a DNSSEC measurement except for PowerDNS which is a regular DNS measurement.

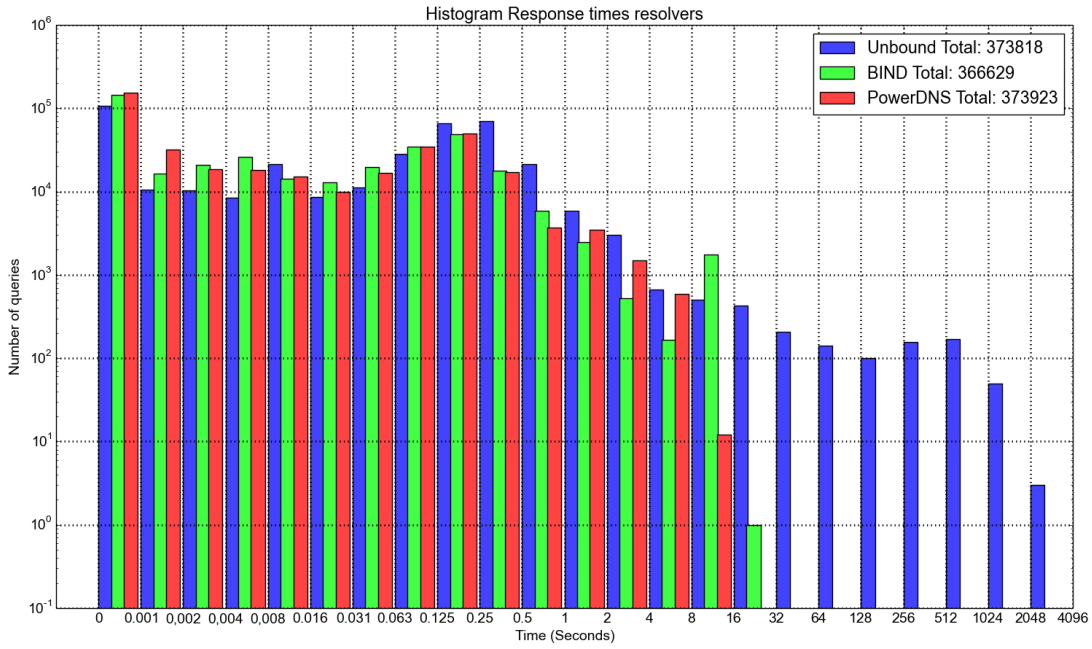

Figure 4.6: DNSSEC Latency Measurements

These results show similarities with the measurements shown in figure 4.2. These similarities are especially seen when it comes to the latencies between 0 and 1 ms which are likely responses from cache. Other similarities are the "bump" between 63 and 500 ms and the different timers of the resolvers which can be seen at the right side of the histogram. Despite the similarities there are some differences in the plot, which are discussed in the following sections.

4.2.1 Unbound

Unbound is one of the resolvers that is capable of performing DNSSEC validation. The measurements of Unbound with DNSSEC show similar results compared to DNS, although not completely the same. The minor differences could either come from DNSSEC or the Internet. DNSSEC could increase the time a query needs to resolve but it is not clear

whether this increase is visible on the histogram because of the low amount of DNSSEC capable domain names. The reason for this is that the differences could also be a result of the changing environment on the Internet. Because not much can be said about the overall measurements between DNS and DNSSEC, the DNSSEC validated results are extracted and shown in figure 4.7.

Figure 4.7 show only the DNSSEC validated data which is 16271 of the total dataset of 373923.

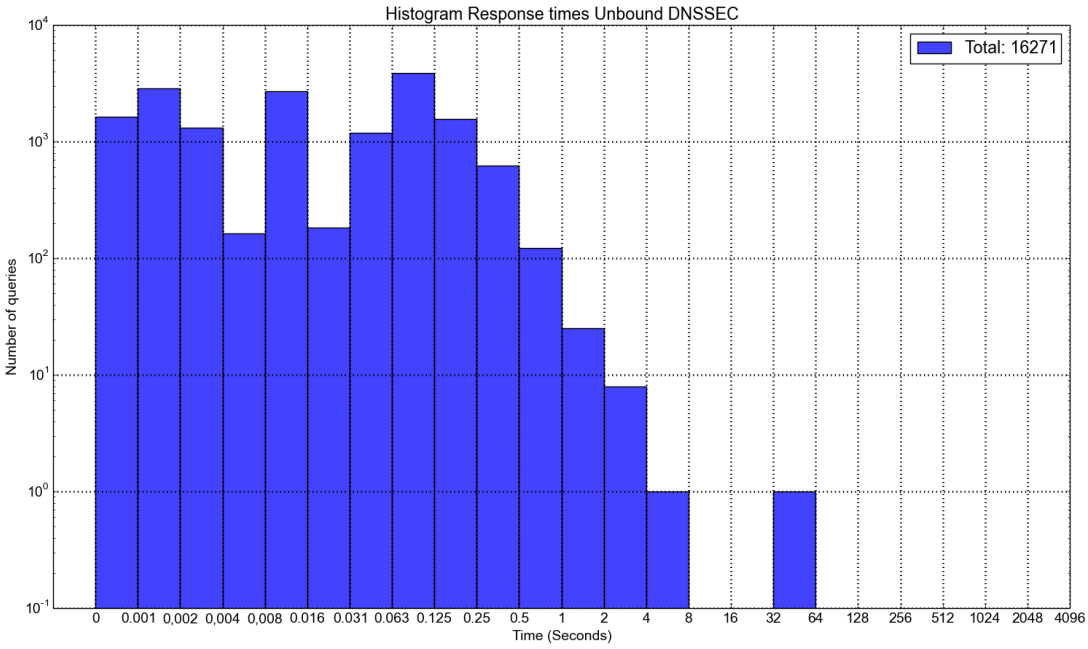

Figure 4.7: Unbound DNSSEC Latency Measurements

As can be seen from the results in figure 4.7 quite some queries are resolved from the cache but also nearby datacenters. This is quite on the low side compared to BIND. A peak is also seen between 8 and 16 ms which is clearly visible as opposed to the overall measurements in figure 4.6. These are likely to be from a country that is close to the Netherlands with quite some DNSSEC capable domains. Another peak occurs between 31 and 250 ms. These are DNSSEC domains that are retrieved from more distant countries, more likely inter continental. It appears that for DNSSEC, Unbound also chooses more distant DNS authoritative servers.

What is also interesting to see is that Unbound is capable of validating domain names even

after 32 seconds due to its repeated trying to resolve queries.

4.2.2 BIND

Measurements have also been performed for BIND, which also supports DNSSEC as described earlier. The data that DNSSEC validated is extracted from figure 4.6 and shown in figure 4.8.

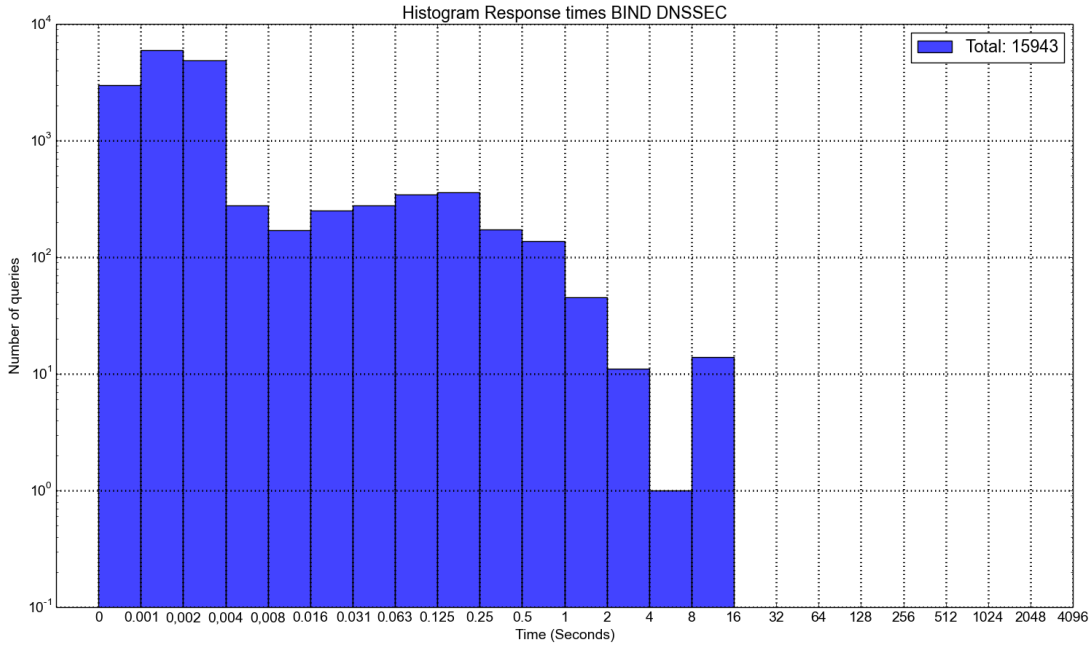

Figure 4.8: BIND DNSSEC Latency Measurements

When looking at the histogram it is clear that it differs for the most part with Unbound. Between 0 and 4 ms it appears that BIND is capable of resolving more from cache than Unbound. But BIND is also capable of contacting more closeby DNS servers to resolve queries than Unbound.

BIND also DNSSEC validates less domain names than Unbound. The difference between the two is 328, which is surprising. One might think that DNSSEC should always result in the same amount of validated domains. The cause for such differences could come from the fact that BIND does not reply to a lot of queries. It could also be that the difference is made by Unbound who tries to resolve more often after failure and therefore succeeds

more often. A last scenario that could be thought of is that resolvers validate differently and could cause a resolver to validate a particular query while another resolver detects a problem in the validating process and therefore does not validate.

4.3 Corner Cases

This section describes the corner cases that are revealed after the measurements performed for Unbound, BIND, and PowerDNS. Corner cases are cases in which various resolvers respond differently to the same query. One might think that this is not possible but only looking at the differences between resolvers shown in section 4.1 and 4.2 one should observe that the result differ to a certain extent. Some resolvers have more or less of certain reply codes than others which indicates such corner cases.

Revealing the corner cases is made easier thanks to the methodology and coding used to perform the measurements and analysis. These allowed to extract the differences between the resolvers more easily, which ensures the detection of the corner cases. After detection, corner cases are confirmed and analysed.

The corner cases that are observed during the analysis of the results are mostly no error no data versus servfail. Where one or two resolvers give a no error no data response, the remaining resolver(s) give a servfail. Other, less common corner cases are no error versus servfail.

It should be clear that there is a big difference between an no error (no data) and a servfail. A no error is an answer where the domain names exists and an answer is available. The no error no data on the other side is a case where the queried domain name also exists but not for the requested type. For a servfail a response could not be acquired or the response does not conform to what the resolver accepts.

It is remarkable that in quite some cases DNS resolvers reply different to the same query. As noted earlier the no error no data versus the servfail is one that is seen more often. However, these differences are not always seen at the same resolvers.

PowerDNS versus Unbound and BIND

A number of cases have been found where PowerDNS replies with a servfail to a query and Unbound and BIND reply with a no error no data. Although this research primarily focusses on revealing the corner cases, a description of the observation is given.

More intensive research is performed to understand why these resolvers give a different answer. By analysing the packets that flow between the resolver and authoritative server, one tries to understand matters better.

In the first place PowerDNS appeared as if it did not interpret no error no data correctly. To confirm or reject this scenario, traffic was analysed carefully.

The traffic analysis revealed that in the most cases the resolvers contact the same authoritative server to answer the query. They also appear to receive the same response from these servers. Yet, the resolvers present different responses to the end-user. The (supposed) authoritative server for these domains give a no error no data response to the DNS resolvers. It is therefore remarkable that PowerDNS gives a servfail to its end-users.

An important detail that could be the cause of these differences between resolvers is the authoritative answer flag in the DNS response. What these corner cases have in common is that they do not have the authoritative answer flag set, while normally it is. If this is the cause of this corner case, this means that PowerDNS does not accept non-authoritative answers and BIND and Unbound do accept this.

This would also mean that for this corner case PowerDNS is the one that is strict while the others are more lenient because of accepting non-authoritative answers.

BIND versus Unbound and PowerDNS

Other corner cases that have been found are cases where BIND replies with a servfail to a query and Unbound and PowerDNS reply with no error no data.

The responses that are given from the authoritative servers are identical over the three resolvers. A closer look at the responses immediately reveal that there is something wrong with the reply that is given from the authoritative DNS servers to the resolvers. The replies which are from domains a couple of levels deep in the DNS tree contain SOA records that have a dot (.) as their names. The dot is of course only used at the DNS root servers and therefore does not belong in these DNS servers.

Even though the dots are present in the SOA records in domains that should not have them, the DNS reply as a whole is a legit DNS reply. It appears that BIND does not accept such a reply and therefore returns a servfail to the end-user. The remaining resolvers both give a no error no data but differ slightly in the answer they present. PowerDNS even returns the incorrect SOA record with the dot as the domain name to the end-user. Unbound lies in between when it comes to this corner case, because is does give a no error no data as PowerDNS, but it does not present the incorrect SOA record to the end-user. BIND appears to be strict when it comes to this case while PowerDNS is lenient. Unbound lies in between the two.

Unbound versus BIND and PowerDNS

A single corner case has been found where BIND and PowerDNS give a no error with an answer and Unbound on the other side gives a servfail for the same query.

Notable was that there were 10 CNAMEs to get to the answer of the A record which is

uncommon but possible. PowerDNS and BIND do not seem to have a problem with this amount of subsequent CNAMEs. Unbound appears to have a limit of eight subsequent CNAMEs and therefore responds with a servfail.

Unbound applies restrictions here and the other do not really care how many CNAMEs they have to follow before an answer is acquired. This does not necessarily have to do with strictness or leniency of resolvers. But it does show that Unbound thought of these cases and applied restrictions.

Unbound versus PowerDNS versus BIND

Another corner case that is revealed gives a no error to Unbound, servfail to BIND and servfail or no error to PowerDNS. PowerDNS strangely tends to vary in response over time while the others have a deterministic answer.

After further analysis it appears that the authoritative server that serves the domain name has four name servers, which is absolutely fine. Surprisingly one of these servers is not reachable for some reason, which is the one PowerDNS more often tries to contact to resolve the query. This therefore gives more often a servfail than a no error reply.

It is not clear what BIND actually does even after analysing the packets. It therefore needs further attention to uncover the cause of this corner case. It is clear that there is something happening with the authoritative name servers which not all have an answer to the query.

Unbound, the opposite of BIND for this corner case is able to resolve this query. There was not a single time it was not able to. After analysis of traffic between Unbound and the authoritative servers, it became clear that Unbound also contacted the authoritative server that was unresponsive. But Unbound showed to also try other authoritative server and was therefore capable of resolving this query.

Quite some corner cases are not directly problems that resolvers cause but rather problems of authoritative servers. However, it is the task of a resolver to do its best to resolve a query. But also, a line must be drawn somewhere for the resolvers, because authoritative servers should also have a responsibility of providing correct data.

Chapter 5

Conclusion

DNS resolvers fulfil an important role in the daily use of the Internet. Different implementations of such resolvers have been developed by different organizations. Although performing the same role, DNS resolvers behave differently in terms of characteristics. This report focussed on the performance of these resolvers.

In order to measure performance of DNS resolvers, a methodology is devised that is able to do this in an objective manner. This also allowed various resolvers to be compared and analysed. The methodology uses the Internet and query traces to prevent biased measurements.

The main research question for this project is, "What is the performance of various DNS resolver implementations?"

The analysis of the performance of DNS resolver implementations has revealed quite a lot on their behaviour. PowerDNS shows to be a resolver that is about getting a response as quick as possible. It almost seems that this implementation prefers a "failed response over a late response". On the opposite, one has Unbound, which is a resolver that besides getting a response a quick as possible also has the motto of "an answer is better than no answer, no matter how long it takes". BIND is a resolver that is in between the two but leaning more towards the behaviour of PowerDNS. But instead of the "failed response" for PowerDNS, BIND tends to be a "no response over a late response" resolver, because quite some responses were not returned.

Also part of the analysis is the unveiling of corner cases in DNS resolver implementations. The devised methodology allowed these corner cases to be found, where after these are analysed. It appears that DNS resolvers do not only differ in performance but also in the response they give to certain queries. A number of corner cases have been revealed but need further analysis. What can be concluded is that certain resolvers behave either more lenient or strict when it comes to accepting and sending DNS data. For the corner cases described in this report, for certain corner cases BIND is more strict while in other cases PowerDNS appears to be really strict. Unbound is more of a resolver that is lenient but not too lenient.

Chapter 6

Future work

DNS resolver measurements is a ongoing process where measurements should be performed continuously. This is due to the protocol where improvement is an ongoing task, new functionalities are added and software releases also keep coming.

Tweaking the resolver configuration for optimal performance (for a certain environment) to perform measurements was not part of this project. It would be interesting and of added value to see how resolvers perform when configuration is changed. There might be a difference in performance between the resolvers when certain configuration is applied.

In order to gain different insights in performance of DNS resolvers, devising other methodologies is encouraged. Measuring with other datasets than the one used in this project could also help in gaining different insight on DNS resolvers.

Another important aspect are corner cases. A number of corner cases have been found as described in section 4.3. It is believed that a certain degree of corner cases exist in resolvers, and it is important that as much as possible are uncovered. This allows one to understand DNS resolvers better and also improve them. The ones that are uncovered in this report need further attention in terms of analysis and possibly modification of resolvers.

References

[1] B. Ager, W. Mühlbauer, G. Smaragdakis, and S. Uhlig. Comparing dns resolvers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 15–21, New York, NY, USA, 2010. ACM.

[2] AppNeta. Tcpreplay website. `http://tcpreplay.appneta.com/` Accessed on: 2015-07-07.

[3] CZ.NIC. DNS authoritative server benchmark measurements. `https://www.knot-dns.cz/pages/benchmark.html` Accessed on: 2015-07-07.

[4] J. Hunter. Matplotlib website. `http://matplotlib.org/` Accessed on: 2015-07-07.

[5] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. Dns performance and the effectiveness of caching. *IEEE/ACM Trans. Netw.*, 10(5):589–603, Oct. 2002.

[6] O. M. Kolkman. Measuring the resource requirements of dnssec. 2005.

[7] N. Labs. DNS authoritative server performance measurements. `http://www.nlnetlabs.nl/blog/2013/07/05/nsd4-performance-measurements/` Accessed on: 2015-07-07.

[8] T. . Libpcap. Tcpdump website. `http://www.tcpdump.org/` Accessed on: 2015-07-07.

[9] Nominum. Nominum website. `http://nominum.com/measurement-tools/` Accessed on: 2015-07-07.

[10] Y. Sekiya, K. Cho, A. Kato, and J. Murai. Research of method for dns performance measurement and evaluation based on benchmark dns servers. *Electronics and Communications in Japan (Part I: Communications)*, 89(10):66–75, 2006.

[11] D. Song. DPKT website. `https://github.com/kbandla/dpkt` Accessed on: 2015-07-07.

[12] W. C. Wijngaards and B. J. Overeinder. Securing dns: Extending dns servers with a dnssec validator. *IEEE Security Privacy*, 7(5):36–43, 2009.

Appendix A

Python code for Analysis

Python code is created for the purpose of data extraction and visualisation of DNS pcap files. This code, which is used for analysis can be found at: `https://github.com/hboulakhrif/ResolverAnalyzer`