You are the weakest link - identifying single-rack points of failure in the DNS

Konstantin Averkin
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
k.averkin@student.utwente.nl

ABSTRACT

The Domain Name System (DNS) plays a vital role in accessing content online as it removes the need for the end-user to remember complicated IP addresses. If this system is unavailable, domain names and other services like email are unusable. In other words, if DNS is down - the Internet is down. In this research, we designed a methodology that identifies single-rack-points of failure in DNS. Natural disasters, human errors, or even cyber-attacks could be the cause of single-rack points of failure in the DNS. To understand if this issue is common, we perform latency and Traceroute measurements with RIPE Atlas against a series of authoritative-name servers which we retrieved from the OpenINTEL active DNS measurement system. For more reliable measurements, with the help of the Haversine formula, we have incorporated the selection of close target probes into the methodology. We find that around 17% of all tested sets of DNS servers are vulnerable to the single-rack point of failure. When comparing the two methodologies with each other we noticed that latency measurements are less reliable than Traceroute. Furthermore, among all tested domain names relying on vulnerable DNS servers, we found that around 11% are vulnerable. In addition, the most affected top-level domains include countries like Russia, Iran, Turkey, Algeria, Japan, and Italy. On the other hand, the least affected TLDs are ".com", ".net" and ".org".

Keywords

DNS, latency measurements, single-rack points of failure, RIPE Atlas.

1. INTRODUCTION

The Domain Name System (DNS) acts as a phone book to translate domain names into IP addresses. An end-user enters a domain name into a browser to access content online. This system eliminates the need to memorize IP addresses and helps to reach online resources instantly. However, if DNS becomes unavailable due to a natural disaster, DDoS attack, or any other catastrophe, the content online is no longer available to the end-user. According to RFC2182 [6], network operators should place all secondary DNS servers in different geographical locations to prevent single-point-of-failure. In case a link to one of the server breaks, others are then available and can provide all of the necessary functionality. Regarding this, in 2003 an unfortunate event happened in one of the University of Twente buildings. An employee of the university has intentionally set the core of the network and the data centre [19] on fire. All of the universitys online resources were no longer available. This has left the whole campus and other services that are dependent on the university's DNS - offline. Luckily, there was a back-up available with all of the necessary information. The service was up and running in a few hours after the incident.

Figure 1: An example of a server rack [5].

Performing this research on the resiliency of DNS servers allows understanding how common it is for domain name owners to rely on the same-rack name servers. A rack server is a computer situated in a rectangular structure which is known as server rack [5]. An example of a server rack can be seen in Figure 1. Having many servers in a single rack makes it easier for the technicians to maintain the rack and has many advantages over performance. With a world-scale measurement platform like RIPE Atlas [18] it is possible to analyze latency and network route between probes and a series of DNS servers. To speed-up the research, we have obtained pre-filtered data from the OpenINTEL measurement platform. The data contains possible candidate pairs of single-rack servers. This platform actively captures daily snapshots of a large part of the DNS, providing DNS operators and academic researchers with vital information [12].

This paper focuses on designing a methodology that iden-

34th Twente Student Conference on IT Jan. 29th, 2021, Enschede, The Netherlands.

tifies whether two given authoritative-name servers are in the same rack.

Next to the main problem, a few follow-up research questions are defined:

1. How reliable are the ways to determine if two targets are in the same-rack?
2. How many domain names are relying on the single-rack of failure DNS servers?

The rest of the paper is structured as follows: Section II describes the tools used for the measurements; Section III methodological approach is described in-depth; Section IV presents the findings for each research question; Section V presents discussion; Section VI limitation and future work; finally, Section VII concludes the paper.

2. METHODOLOGY

This paper focuses on designing a methodology given *Ping* and *Traceroute* measurements. The methodology identifies whether the domain names are vulnerable to the DNS single-rack points of failure.

2.1 Probe selection

To achieve accurate measurement results we have implemented a separate process for selecting probes. The closer probes are to the target - the more consistent results are achieved. This is a crucial part of this methodology. Probes that are closest to the target provide the most accurate results. This is because there are fewer intermediate machines thus, less traffic in the path. Due to RIPE Atlas credits limitations, a single user is only allowed to use at most 1000 probes. We are using the Haversine formula to estimate the distance from a probe to a target (see Equation 1). With this formula, it is possible to determine the great-circle distance given a pair of latitude and longitude points. The small r denotes the distance in kilometres around the equator. The symbols ϕ and λ represent latitude and longitude pairs. Once the distances between probes and the target are known, we are sorting them in ascending order. Then we select the first 1000 closest probes to the target for the RIPE Atlas measurements.

$$d = 2r \arcsin\left(\sqrt{\sin^2(\tfrac{\phi_2-\phi_1}{2}) + \cos(\phi_1)\cos(\phi_2)\sin^2(\tfrac{\lambda_2-\lambda_1}{2})}\right) \quad (1)$$

2.2 Ping

The *Ping* measurements are simply the time periods during which an echo request message traveled by using the Internet Control Message Protocol from the host to the target and back. The whole amount of time needed from sending to receiving the packages for the host is also known as Round Trip Time (RTT). The Ping measurements allow this research to understand the closeness between the two DNS servers. Due to the fact that not all the measurements are successful, we filtered out some entries of the data. Once the datasets are ready, it is possible to start to design the methodology. The basic idea is to calculate the average RTT distances between two targets for the closest 1000 probes. For each measurement, the methodology sends a total of 10 packets from the host to the target. The measurements then return the average RTTs for each probe. The designed methodology calculates the differences probe-by-probe between two targets. Then we sort the differences in ascending order (they can be either positive or negative) and plot them in a graph. In theory, the differences are close to zero only if two of the measurement's average RTTs are close to each other. If probes have similar latency between two targets it is safe to assume that the targets are close to each other. Another indication is to look at where the middle point or the lowest difference value is positioned. If the middle point is centred, the differences balance each other out. If they are skewed either to the right or left, the average differences are imbalanced. Consequently, the target's physical positions differ. Ideally, if two servers are in the same rack, the first and the second 50% of the probe's differences are equally distributed. This would mean that the mean point of all the probes is placed exactly at 50% mark. In other words, if dispersion (difference from the ideal middle point to the measured) is narrow it is considered that servers are in the same rack. However, the most challenging part of this methodology is to define the threshold on acceptable dispersion. These thresholds are the core of this methodology since they identify whether the two servers are in the same rack. The test experiment explains the way we select the thresholds.

2.3 Traceroute

Understanding if two targets share the same route to a probe provides a better understanding of the closeness. Traceroute's measurements explain crucial details about intermediate nodes between a host and a target. If two or more servers are connected to the same router, then we consider that they are in the same rack. To put this in practice, we designed a methodology that checks the last hops in the route. The more matching last nodes there are between the paths - the more likely it is that the two targets are in the same rack. In the end, the methodology counts each of the entries and estimates a usage percentage for the most common node. Having an understanding of the percentage of probes that are using a certain so-called hop, it is possible to start defining thresholds. Once we compare results to the ground truth of certain targets we start analysing possible constraints. Based on the observations of a test experiment, we define thresholds for the Traceroute methodology. If there are more than two DNS servers in a single set, we compare the last hop nodes with every target's path within a set. If all of the most common nodes match, all authoritative-names servers are considered to be in the same rack.

2.4 Measurement tools

The most used platform for this research is RIPE Atlas, which currently has 11394 connected probes all around the world [16]. Probes are a piece of hardware that are hosted by any end-user with a stable network connection either at home or in the office. There are no requirements to receive and host a probe. RIPE Atlas uses a credit-based system to allow researchers to use the platform, perform measurements, and avoid abuse. Researchers can gain these credits by hosting probes and then use them to perform their own network measurements. In this research, we will use the credits provided by the RIPE Atlas community.

We have retrieved the data from the OpenINTEL database. This platform actively performs global DNS measurements and provides this valuable information to the research community [21]. Currently, it has collected 4.8 trillion data points, and it has been swiftly growing since 2015 [12].

We have written the methodology in the Python programming language as there are many libraries conveniently defined. In addition, a Python framework is required to be able to communicate with RIPE Atlas REST API. Python libraries such as Cousteau [3] allows for performing a new or retrieve any available RIPE Atlas probe measurements.

We are using other libraries like Pandas [14], Numpy [11] or PythonPing [15] to help process and analyze the data. Data visualization libraries like Matplotlib [10] allow for seeing data patterns and determining thresholds.

2.5 Test experiment

In order to define certain thresholds for Ping and Traceroute methodologies, we have conducted a test experiment with a small set of targets. In total, we have used 5 targets of which 4 were known to be in the same rack. The same-rack targets are named A, B, C, and D. The last target is close to the other probe, however, not in the same rack. From now on this different rack-server is called E.

The first part of the experiment is performed as follows.

1. Check if all the targets are reachable. In order to do so, we send 4 Ping requests per each target in a set. This helps to reduce the usage of credits when we perform RIPE Atlas measurements.

2. If at least 3 Ping requests are successful for all targets then we assume that all the targets from a set are reachable. The next step is to use the Haversine formula and select the closest probes for each target. This allows us to have the most accurate measurement results which we then use in the following experiment.

3. Then we send Ping and Traceroute measurement requests against series reachable targets to the RIPE Atlas platform. If they are successful, we locally organise measurement IDs and save them to a file.

The second part is summarized as follows.

1. Parse the measurement IDs and start the above described Ping and Traceroute methodologies. Gather the results.

2. Then we analyse the results and choose relevant thresholds. We determine these thresholds by looking at the differences between the same and not the same rack machines.

From this small experiment, results are summarised as follows:
For Ping methodology, if the middle point is variating less than 10% to either left or right, we can assume that two DNS servers are in the same rack. For example, comparing the 4 same-rack machines (A, B, C, and D) resulted in the middle points between 41% and 52% (see Figure 2). While comparing same-rack machines with E, resulted in middle point variating between 30% to 35%. We see that points are close to the 40% threshold. However, the pre-defined ground truth states that the E machine is in a different rack. The above-mentioned thresholds are only taken into consideration if more than 100 pairs of probes are participating in the measurement. Else, the set of DNS servers is considered to be on different racks. Moreover, the red line represents the differences in averages between same-rack targets B and C. While black line represents average latency differences between same-rack machine D and different rack target E. These lines signal the reasoning of middle point skewness. For example, the line can explain why the black target has wider dispersion compared to the same rack machines. It can be seen that the black line has high average latency differences (more than 20ms) around 0% to 15% mark of probes comparing it to the red. Furthermore, a similar effect can be seen for the red line. Around the mark of 90%-100% differences go above the 20 milliseconds. For Traceroute methodology, if more than 60% of probes are using the same last hop, then the two targets are in the same rack. In this experiment, more than 90% of probes using the same route to the targets in the same-rack (see Table 1). The methodology only considers measuring further if there are more than 100 pairs found. Else, we consider that the set of DNS servers in different racks. When compared with the other machine, there were almost no pairs on the same path.

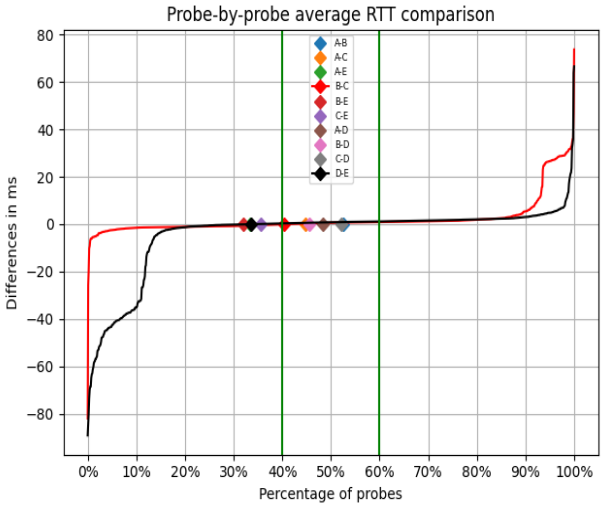

Figure 2: Average probe-by-probe latency comparison between same and different rack machines. Targets A, B, C and D are known to be in the same rack. While target E is close to the other for however, in different rack.

Table 1: Traceroute probe-by-probe last hop comparison.

Pairs	# most common	Total pairs	Percentage
A - B	926	933	99.25%
A - C	923	938	98.40%
A - D	924	934	98.93%
B - C	925	932	99.25%
B - D	926	937	98.83%
C - D	923	933	98.93%
A - E	3	8	37.5%
B - E	3	5	60.0%
C - E	3	8	37.5%
D - E	3	5	60.0%

In order to further test the methodologies, we have measured around 587 DNS servers. The methodology uses the pre-defined thresholds this time. In the following section we describe the data organisation.

2.6 Data organisation

To perform the measurements, the data is organized in a structured way. In total two datasets are used.

1. A subset of entries collected from Alexa top one million datasets. It contains around 20 000 sets of DNS IP addresses. Next to the addresses, the corresponding domain names are stored. Furthermore, in order to use the Haversine formula a target's geolocation data is required. This includes the country code and coordinates data *latitude* and *longitude*.

2. Another dataset needed for this research is information about the probes. We have parsed the information about probes from the official website of RIPE NCC [17]. We have used the probes that were active during the day of January 9th, 2021. In general, we have used only *ID*, *IPv4 address*, *latitude*, *longitude* and *status name*. The geolocation data is needed to calculate the distances between probes and targets. Once we know the distances, the methodology uses relevant probe IDs are as a source of the measurement.

3. RESULTS

This section describes the output of each methodology and tries to answer the following research questions. In total we have tested 587 sets of which 102 were vulnerable. The results of both methodologies are visible in the Figure 3.

3.1 How reliable are the ways to determine if two targets are in the same-rack?

As it turns out, the Ping methodology results show that more than 80% of sets are in the same rack. The Ping methodology had multiple times higher acceptance rate than Traceroute. Thus, resulting in only 9.2% rejection rate. Some measurement results are unusable due to the small number of probes participating. To be more concrete, 13.3% of Ping methodology results reported being unusable.

On the other hand, the Traceroute results acted more like an extension towards identifying same-rack DNS servers. Out of all Traceroute methodology results, we have identified that around 90% of sets were as in the same rack compared to the Ping methodology. The acceptance rate for this methodology resulted in 19.25%. Comparing the acceptance rate to the before-mentioned Ping methodology this is a moderate decrease by 60.75%. Consequently, the rejection rate has increased to 55%. Many of the results (323) were not accepted due to the threshold. Furthermore, more than 25% of tested Traceroute sets showed no results.

Ping methodology is not as reliable as Traceroute. However, the combination of two narrows the results to a lower acceptance rate. The combination of both methodologies shows that 102 sets of DNS servers are vulnerable out of a total of 587. The acceptance rate for this methodology is now only 17% of all sets. The combined methodology rejects a few more sets when comparing the results to Traceroute's methodology. The rejection rate has increased by 1.5% and finally resulted in 56.56%. The amount of sets that showed no results has resulted in the same number as in Traceroute.

3.2 How many domain names are relying on the single-rack of failure DNS servers?

There were a total of 3557 domain names relying on 587 sets of DNS servers. The combination of the methodologies confirmed that 102 sets of domain name servers are vulnerable. Consequently, 394 domain names are relying on vulnerable DNS servers according to methodologies. This is around 11.1% of tested domain names.

Table 2 represents the most affected top-level domains. In total there are 44 different top-level domains affected out of 168 tested. 75% of all tested domain names had the top-level domain extension ".com" however, only 135 domains are indeed vulnerable to the single-rack point of failure. The most vulnerable top-level domains appear to be Russia's ".ru" 117 out of 156 and Iran's ".ir" 29 out of 85. There are a few more affected top-level domains worth mentioning. For example, Turkey's ".tr" top-level domain has 12 out of 28 tested domain names relying on vulnerable DNS servers. On the other hand, generic top-level domains such as ".net" or ".org" have a relatively small number of affected domain names. Only 15 out of 156 and 9 out of 124 have been affected, respectively. Also, the France top-level domain ".fr" has only been affected by 8 out of 41 domain names. Other, less common TLDs were Algeria's ".dz" of which all tested domain names were vulnerable, Japan's ".jp" 5 out of 9, and Italy's ".it" with 4 out of 10.

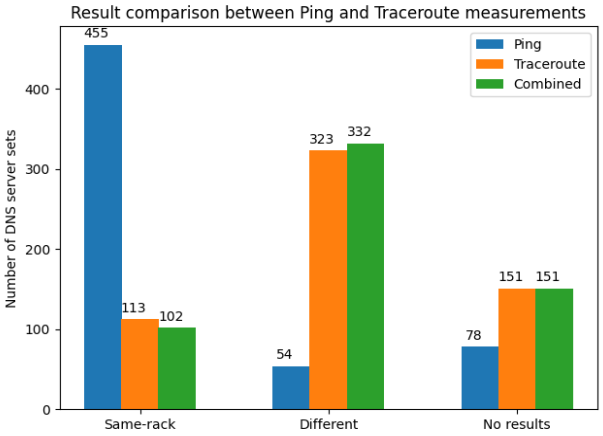

Figure 3: Ping, Traceroute and combined results comparison.

Table 2: Top 10 most affected top-level domains.

TLD	# of affected	Total	Percentage
.com	135	1859	7.26%
.ru	117	156	75%
.ir	29	85	34.12%
.net	15	156	9.62%
.tr	12	28	42.86%
.org	9	124	7.26%
.dz	9	9	100%
.fr	8	41	19.5%
.jp	5	9	55.56%
.it	4	10	40%

4. RELATED WORD

In the past, there has been some research done regarding estimating physical locations by performing latency measurements. However, there has been little research on estimating the closeness between two separate servers. Nevertheless, the preparation of measurement data is rather similar.

4.1 Probe selection

To have stable measurement results, we are focusing on selecting probes that are as close as possible to the target. Since latency highly depends on the number of intermediate nodes, the methodology needs to take into account the delay of the hardware processing. Thus, the more hops in the path, the higher the delay [1]. Another reason why selecting all probes for better coverage does not work is because it is not scalable well in practice [2]. Be-

sides the closest probe selection mechanism, there are a few other approaches. For example, [23] introduces a linear programming model to select relevant landmarks used for measurements. However, the approach seems to be irrelevant when considering measurements on a large scale. Furthermore, the other paper [8] discusses that to achieve the most accurate results, the methodology should select the closest probes. The reason for this is that there is the lowest interference between the target and the landmark.

4.2 Identifying physical locations

In this research, we focus on identifying whether two DNS servers are in the same physical location. There has been some similar research on this problem however, the majority of the related work papers focus on estimating exact physical location. There are several ways of estimating the approximate target location. Because the host landmarks have a fixed physical location, the simplest way of finding out where the target resides is by choosing the lowest latency value [13]. In practice, the shortest Ping results are comparable to more complex algorithms [9]. However, in this research, we focus on understanding whether two servers are likely to reside in a similar location. Thus, this methodology uses an average of 10 Ping measurements and compared probe-by-probe.

There exist some other delay-based geographical location techniques according to Katz et al, for example introducing the global delay-to-distance conversion method by using the speed of the Internet. A speed of Internet constraint is used to convert delay-to-distance. A similar technique could be used to understand if two servers are likely to reside in the same physical location. However, this problem requires further research.

A few other possible solutions for estimating targets location in places with high traffic could be to use network coordinates [4] or network location services [22]. Both of these methods initially have a different approach. Dabek et al [4] have designed an algorithm such that each host computes its own coordinates by performing latency measurements with other machines. Wong et al have performed [22] similar research by first performing node selection and then trying to Ping targets from them. Then, the methodology selects the closest probes with the help of latency thresholds. Such a methodology allows to perform measurements more efficiently and even have a higher accuracy rate. Part of the idea from [22] is implemented in this paper's probe selection mechanism. Except we do not create an algorithm for discovering closest probes, we estimate the distances and only pick the closest probes. Other, more complex solutions are described in [20], where various hyperspace distance functions were investigated by using *Ping* and *Traceroute*. In the researcher, performed by van Langen et al, multiple hyperspace functions are studied against the actual distance between hosts. The best outcome for *Ping* is observed with a lower-bound distance function and for *Traceroute* the average of the sum of lower and higher bounds.

In this research, we used Traceroute measurements to compare the last hops between two paths from the same probes to different targets. If the last hops match, for more than 60% of landmarks, we consider that targets are in the same rack. Other researches [7] suggest checking if two paths have the same or similar amount of hops in the path. If the difference is low, try to locate the closest node and repeat Ping measurements to it. This certainly reduces accuracy however, gives for the methodology some indication if two servers are close to each other.

5. LIMITATIONS AND FUTURE WORK

5.1 Limitations

Ping and Traceroute closeness measurements highly depend on the number of participating probes. The more probes are participating in the measurements, the more accurate results the methodology produces. Due to RIPE Atlas credit limitations, the methodology can only use 1000 probes per measurement. Furthermore, other credit limitations are also influencing the number of DNS sets we test. Daily credit limitations allowed to only test at most 30 sets of DNS servers per day.

Another limitation is that the Ping methodology's thresholds were too wide. Such behaviour could be explained by the global threshold not being applicable for each set of domain name servers. The latency averages tend to deviate more due to the small concentration of probes around the target. Thus, the measurements are less reliable and the threshold needs to be adjusted individually. Furthermore, to test the methodologies we have used a snapshot of connected probes during January 9th, 2021. However, not all probes have 100% uptime and some are disconnected. Thus, to further improve the accuracy of the research, new snapshots should be parsed from official sources each day new measurements are performed.

Additionally, because not all probes are concentrated equally around the globe, there is some room for improvement for probe selection mechanism. Instead of selecting a certain amount of the closest probes, it would be better to decide on the distance surrounding a target. Consequently, this would lead to measuring with a lower amount of probes in less concentrated areas. However, this would improve measurement accuracy when designing a methodology using global thresholds.

5.2 Future work

To further improve the research, it could be possible to test the results with some ground truth. For example, reaching out to operators which maintain our identified single-rack DNS servers. Perhaps they would be able to answer the question of whether servers are indeed in the same rack. Moreover, if servers are in the same rack, the reasons why servers are operating against the recommendations could be discussed.

In the process of this research, we noticed that same-rack identification would be more reliable if the Ping methodology would be an extension for Traceroute instead of another way around. This would be a more efficient way of measuring and would require fewer credits for the RIPE Atlas platform. Thus, allowing to find more single-rack points of failure.

Another improvement is to filter out unreachable targets with a higher accuracy rate. In this research, from the Traceroute results, many targets were unreachable. This is because intermediate nodes do not allow to report the state of the machine. A better target selection is needed to tackle this problem. Instead of only performing Ping measurements from the researcher's computer, we could also do initial Traceroute measurement against the targets. This would reduce the number of sets that are usable in the research.

The results of the second research question provoke some further research. We noticed that the majority of affected top-level domains are based in countries with lower economical and technological development. One of the possibilities would be to analyse the relationship between countries with lower than average economic status and single-

point of failure in the DNS. It is expected that such countries are more likely to suffer from this vulnerability.

6. CONCLUSION

In this research, we have designed Ping and Traceroute methodologies that identify whether two given DNS servers are in the same-rack. To test the outcome of both methodologies, we performed an experiment against a series of authoritative-name servers. Initially, a smaller scale experiment is performed to adjust the methodologies according to the given ground truth. Besides designing a methodology, we answered two follow-up research questions and summarized them as follows.

The performed experiment concluded that the combination of two methodologies outputs the most reasonable results. The latency methodology requires additional testing to further analyse single-points of failure in the DNS. The defined thresholds in the Ping methodology have a high acceptance rate. This was caused due to variating probe environment, inconsistent measurement results, and the thresholds. On the other hand, the Traceroute methodology successfully identified whether targets are connected through identical gateways. The acceptance rate for this methodology was twice lower compared to Ping. Combing both methodologies results showed that out of the total of 587 tested sets of DNS servers, 102 were identified as vulnerable to the single-rack point of failure. Overall, there is room for improvement for both methodologies however, current results indicate that Traceroute methodology outputs more reliable results than Ping.

Once we designed the methodologies, it is possible to quantify the domain names vulnerable to the single point of failure. We tested a total of 3557 domains and found 394 to be vulnerable. Even though this is the only 11% of all tested domains, it is clear that this problem is crucial to the owners of the domain names. Furthermore, we have analysed a set of affected top-level domains. As it turns out, the most affected TLDs were based in Russia, Iran, and Turkey. Other less common however also affected were country TLDs from Algeria, Japan, and Italy. The most tested and least affected were the most used top-level domain ".com" and two generic top-level domains ".net" and ".org".

In conclusion, the results presented that single-rack point of failure is common among 11% of tested entries. The methodologies have room for improvement however, the combination of both methodologies identifies whether two given servers are in the same physical location. Latency and Traceroute measurements with platforms such as RIPE Atlas provides insightful information about each target and the path in-between.

7. REFERENCES

[1] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Van Mieghem. Analysis of end-to-end delay measurements in internet. In *Proc. of the Passive and Active Measurement Workshop-PAM*, volume 2002. sn, 2002.

[2] M. Candela, E. Gregori, V. Luconi, and A. Vecchio. Using ripe atlas for geolocating ip infrastructure. *IEEE Access*, 7:48816–48829, 2019.

[3] Cousteau. `https://github.com/RIPE-NCC/ripe-atlas-cousteau`. Last accessed: 20.11.2020.

[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. *ACM SIGCOMM Computer Communication Review*, 34(4):15–26, 2004.

[5] B. Daniel. What is a rack server? `https://www.trentonsystems.com/blog/what-is-a-rack-server`. Last accessed 23.01.2021.

[6] R. Elz, R. Bush, S. Bradner, and M. Patton. Selection and operation of secondary dns servers. `https://tools.ietf.org/html/rfc2182`, July 1997. Last accessed: 20.11.2020.

[7] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of internet hosts. *IEEE/ACM Transactions On Networking*, 14(6):1219–1232, 2006.

[8] P. Hillmann, L. Stiemert, G. D. Rodosek, and O. Rose. Modelling of ip geolocation by use of latency measurements. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 173–177. IEEE, 2015.

[9] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards ip geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 71–84, 2006.

[10] Matplotlib. `https://matplotlib.org`. Last accessed: 20.11.2020.

[11] Numpy. `https://numpy.org`. Last accessed: 20.11.2020.

[12] OpenINTEL. `https://openintel.nl/background/`. Last accessed: 20.11.2020.

[13] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for internet hosts. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 173–185, 2001.

[14] Pandas. `https://pandas.pydata.org`. Last accessed: 20.11.2020.

[15] PythonPing. `https://pypi.org/project/pythonping`. Last accessed: 20.11.2020.

[16] RIPE. Ripe atlas. `https://atlas.ripe.net/results/maps/network-coverage/`. Last accessed: 20.11.2020.

[17] RIPENCC. `https://ftp.ripe.net/ripe/atlas/probes/archive/2021/01/`. Last accessed: 9.01.2021.

[18] R. N. Staff. Ripe atlas: A global internet measurement network. *Internet Protocol J.*, 18(3):2–26, 2015.

[19] UTwente. The fire in the tw/rc building. `https://www.utwente.nl/en/organisation/alumni/ut-canon/stories/fire/`, November 2002. Last accessed: 20.11.2020.

[20] S. Van Langen, X. Zhou, and P. Van Mieghem. On the estimation of internet distances using landmarks. In *Proc. of the International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking–NEW2AN04*. Citeseer, 2004.

[21] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras. A high-performance, scalable infrastructure for large-scale active dns measurements. *IEEE Journal on Selected Areas in Communications*, 34(6):1877–1888, 2016.

[22] B. Wong, A. Slivkins, and E. G. Sirer. Meridian: A lightweight network location service without virtual

coordinates. *ACM SIGCOMM Computer Communication Review*, 35(4):85–96, 2005.

[23] A. Ziviani, S. Fdida, J. F. De Rezende, and O. C. M. Duarte. Improving the accuracy of measurement-based geographic location of internet hosts. *Computer Networks*, 47(4):503–523, 2005.