DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Analyzing Cloud Reachability on Global Scale

Maximilian Eder

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Analyzing Cloud Reachability on Global Scale

Analyse der Cloud-Erreichbarkeit im globalen Maßstab

Author:	Maximilian Eder
Supervisor:	Prof. Dr.-Ing. Jörg Ott
Advisor:	Dr. Nitinder Mohan
Submission Date:	14.11.2020

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 14.11.2020 Maximilian Eder

Acknowledgments

I would like to thank my supervisor Prof. Dr.-Ing. Jörg Ott for allowing me to write this thesis at his chair. I also want to thank Dr. Lorenzo Corneo for providing his ERA toolkit and for letting me co-author a paper about my thesis topic. Furthermore, I want to thank my advisor Dr. Nitinder Mohan for his invaluable help in understanding this topic and for motivating me. Finally, I want to thank my friends and family for supporting me throughout my carrier.

Abstract

Cloud computing has seen steady growth over the last decade and its deployment is now increasingly covering the whole planet, while next-gen applications for the cloud come with bigger and bigger requirements. From cloud gaming to autonomous driving and AR, an important metric to enable those applications is cloud access latency. Providers are constantly looking to increase their reach and improve their connectivity, with many negotiating peering agreements directly with ISPs and some even building their own private backbone to constantly improve user access latency and bandwidth. Consequently, the lack of scientific literature about this topic is perplexing, with the latest global cloud latency study dating back to 2010. In this thesis we perform extensive global client-to-cloud latency measurements towards 189 datacenters from all major cloud providers to analyze the impact of those tactics. We conduct our analysis using the well-known measurement platform RIPE Atlas, involving over 8500 probes from all around the world to create a dataset spanning more than 196 million datapoints. We evaluate the suitability of current cloud environments for modern applications such as virtual reality in different regions around the world. We differentiate our findings between the different cloud providers and attempt to quantify the impact that the privatization of the user-to-cloud-path and ISP peering agreements have on them. Our results indicate that the majority of the worlds population can reach the cloud within the threshold needed to perform even most demanding applications. We find, that the development of private backbones by some cloud providers shows clear improvements of connectivity in some, but not all regions. Our research also suggests, that ISP peering agreements rarely improve latency for cloud providers in most regions.

Kurzfassung

Die Verbreitung von Cloud computing wächst seit Jahren stetig und die Datencenter der Betreiber bedecken zunehmend den ganzen Planeten. Unterdes stellen immer mehr Technologien von morgen immer größere Anforderungen. Angefangen bei Cloud Gaming über fahrerloses Fahren bis hin zur Augmentierten Realität, die wichtigste Kenngröße, um diese Anwendungen möglich zu machen, ist die Latenz vom Nutzer zur Cloud. Die Anbieter versuchen kontinuierlich, ihre Verbreitung und Erreichbarkeit zu verbessern, wobei viele mit den Internetanbietern Verträge zur direkten Verbindung, sogenannte Peering Agreements, abschließen. Einige bauen sogar ihr eigenes Netzwerk, um die Latenz für den Nutzer so niedrig und die Bandbreite so hoch wie möglich zu halten. Umso überraschender ist es, dass die wissenschaftliche Literatur in diesem Bereich sehr dürftig ist. Die letzte globale Cloud-Latenz Studie wurde 2010 veröffentlicht. In der vorliegenden Arbeit führen wir weit reichende Latenz-Messungen an 189 Datencentern aller großen Cloud Anbieter durch. Wir verwenden die bekannte Platform RIPE Atlas, um unsere Messungen von mehr als 8500 auf der ganzen Welt verteilten Computern aus zu starten, wobei wir mehr als 196 Millionen Datenpunkte sammeln können. Mit Hilfe dieser Messungen prüfen wir die aktuelle Cloud Infrastruktur auf Eignung für die nächste Generation an Anwendungen und vergleichen die Ergebnisse von verschiedenen Regionen rund um den Globus. Wir unterscheiden dabei die unterschiedlichen Cloud Anbieter und versuchen, die Auswirkungen ihrer Bemühungen um die Privatisierung des Weges vom Nutzer zur Cloud, sowie ihrer Peering Agreements mit Intenetanbietern, zu quantifizieren. Unsere Ergebnisse suggerieren, dass die Mehrheit der Weltbevölkerung die Cloud schnell genug erreichen kann, um selbst anspruchsvollste Anwendungen zu ermöglichen. Wir finden heraus, dass der Ausbau eines privaten Netzwerks klare Latenzverbesserungen für die Anbieter bringt, allerdings nur in bestimmten Regionen. Unsere Forschung zeigt außerdem, dass Peering Agreements mit Internetanbietern in den meisten Regionen selten zu einer Verbesserung der Latenz führen.

Contents

1. Introduction

1.1. Motivation

The cloud has seen steadily rising interest over the last ten years, with several providers emerging and deploying a rapidly growing network of datacenters all around the globe. The high supply, as well as the equally high demand for cloud computing from companies, individuals and scientific organizations alike make this field inherently interesting for studies. Consequently it comes as a surprise, that the last broad scientific study to the best of our knowledge is dated back to 2010 [1]. Back then, Amazon Elastic Compute Cloud (EC2) for example only had four deployments around the world, compared to 21 now [2]. Furthermore, while back then applications for cloud computing were sparse, nowadays, there seems to be an infinite number of next-gen applications, seemingly predestined for the cloud. Some providers have already rolled out cloud gaming, smart homes are already controlled via the cloud, and while autonomous driving, augmented and virtual reality are already worked on, bigger concepts like smart cities are looming on the horizon - all powered by the cloud. But those applications come with steep requirements: augmented and virtual reality for example - when not calculated fast enough - cause motion sickness and dizziness [3, 4]. But while failing these requirements for AR and VR may result in nausea, the consequences of being too slow are much higher for autonomous driving. To be ready for those applications, user-cloud access latency must be reliably below a certain threshold. Given the vital importance this topic has, the lack of up-to-date literature about current cloud infrastructure is perplexing.

In the last decade, cloud deployment has steadily improved, not only in terms of datacenters, but also beyond that: More and more cloud providers are trying to take control over the path from user to cloud, in order to minimize access latencies. They are doing so by building their own private network, interconnecting their datacenters. Those private backbones are then connected to the users directly, by setting up private peering agreements with their ISPs. This expansion of their private networks has allowed cloud providers to bypass huge other internet traffic and connect directly to the user, something unthinkable ten years ago. These developments are shifting paradigms, as well for the cloud computing field, as for the entire hierarchy of the internet [5]. We therefore think an up-to-date study is needed to evaluate and - if possible - quantify the progress made in the field by cloud providers. We also want to explore the paradigm-shifting methods, that cloud providers have used over the past decade to improve their service for the user and to find out, which factors impact user-to-cloud latency.

1.2. Research Questions

RQ1: What are the user access latencies to the current cloud infrastructure across the globe? Latency is one of the defining metrics of user-to-cloud access. It is important in categorizing the connection for certain applications - from the smart city to live cloud gaming and even augmented reality and virtual reality in the cloud - the access latency a user can achieve is the primary way of measuring user-to-cloud connectivity. Given the wildly different quality of network infrastructure, as well as cloud deployment between countries, it is important to take a look at the global scale and to identify and highlight regions of interest.

RQ2: Does the type of underlying networking backbone interconnecting cloud infrastructure impact user cloud access? With the internet growing faster and faster over the last decades, the demand for a high-bandwidth "exclusive lane" has understandably been booming, especially among cloud providers. To optimize their bandwidth and thereby their customer service, bigger cloud providers like Google, Microsoft and Amazon have steadily increased their efforts to privatize their backbone, even trying to get their network as close to the customer as possible. We want to find out, if and how this way of managing traffic in contrast to using the public internet can be measured. We also want to quantify the improvements of those private WANs in terms of improved latency.

RQ3: What is the impact of ISP peering agreements on global user-cloud access? The access to the internet has traditionally been a hierarchical one. Users accessed tier 3 ISPs, who in turn accessed tier 2 ISPs, who then connected to large, world spanning tier 1 ISPs. But recent research suggests, that the internet's hierarchy is starting to crumble [5]. The internet is steadily flattening, meaning corporations - especially cloud providers - and tier 3 ISPs start peering among themselves turning traditional tier 1 and tier 2 ISPs more and more obsolete. Peering is no longer a solely hierarchical domain, which comes with benefits for both Content Delivery Networks, as well as cloud providers. In theory, this gives the cloud providers access to the ISPs users, while decreasing access latencies for them. We aim to find out, whether ISP peering has the effects we expect it to have and what geographical differences exist.

1.3. Contribution

We conducted a cloud analysis study spanning ten cloud providers on six continents and collected around 196 million data points. We show the current state of cloud reachability in terms of latency for continents and countries alike and compare performances of individual cloud providers. We also provide the context of next-generation applications requiring latencies below a certain threshold to classify the current cloud performance in global regions and find, that a majority of the population is able to connect to the cloud fast enough to enable those applications. This thesis compares the accuracy of ICMP and TCP latency measurements and finds, that the latter proves consistently lower than its counterpart. It also

quantifies the degree, to which cloud providers' private WANs have already taken over the path from the user to the cloud.

1.4. Overview

First, we are giving an overview over the topic and the scientific research already conducted in this field in chapter 2. We then describe our own measurements and how we obtained and enhanced our research data in chapter 3. Here we specifically detail our choice of research targets and the data we want to analyze. We detail our findings in chapter 4 and attempt to answer our research questions. Finally we draw a conclusion in chapter 5 and offer an explanation of how to reproduce our results in chapter 6.

2. Background & Related Work

2.1. Cloud Application Latency Requirements

Mohan et al. [4] analyzed the latency of RIPE Atlas probes to cloud datacenters to discuss the usefulness of edge computing in the cloud. In that context, they compare the latencies to landmarks in the area of latency perception to humans. They define three such latency classes:

Motion-To-Photon latency (MTP) is the delay between user inputs and the reaction happening on the computer screen and marks the latency, at which symptoms like motion sickness can occur and lies at ca. 10-20 ms. It is necessary for applications like Virtual Reality (VR) and Augmented Reality (AR) to stay within those latencies. Other applications, that require such low latencies are autonomous driving and to some extend 360-degree video.

Perceivable Latency (PL) is the latency at which a delay between user input and visual response to that input, becomes noticeable to humans. It is roughly estimated to be 100 ms and is important for applications like wearables, gaming and most camera and traffic monitoring.

Human Reaction Time (HRT) is the last category defined in the paper. It is a necessary delay to beat for most real-time applications over the cloud, including some smart home and smart city applications, as well as applications like remote surgery. It is generally set around 250 ms.

We will refer to those classifications in the thesis.

2.2. Global Cloud Provider Reachability

Since the use of cloud computation nowadays is of ever-growing importance for companies, individual users and scientific researchers alike, it has naturally been subject to extensive research and debate. Massive amounts of reports regarding the latency of cloud computing datacenters have been published, most notably the yearly "Thousand Eyes Report" [6]. In the year 2019, it measured cloud-to-user performance from 98 end-user vantage points in Europe, Oceania, Asia, Africa, North and South America to 94 datacenters belonging to 5 different cloud providers (Amazon, Google, Azure, IBM and Alibaba). It found that in comparison to its last year report (2018) cloud performance measurements improve, depending on the provider, by up to 36%. It also monitored steadily growing investments into architecture and

peering in geographical locations, that previously were showing a performance worse by comparison, especially Asia and Africa. It also highlights the growing backbone infrastructure and predicts steadily growing obfuscation of path steering within the networks of those cloud providers. The report shows, how providers with a superior private backbone like Azure offer better performance predictability than cloud providers, who rely more on the public internet. It analyzes the lack of interconnection between certain locations and the impact those have for customers of the cloud provider in that region, e.g. Google's missing connection between India and Europe and the resulting latency penalties. The report also empirically covers the impact ISPs have on the latency of a customer's experience in connecting to the cloud datacenters. It looks specifically into North American ISPs and determines, which ISPs provide the best latency for cloud users in North America.

Li et al. [1] compared cloud providers 10 years ago and were one of the first to do so. They also present their benchmark suite - *CloudCmp* - to compare cloud providers by cost and performance scales. They specifically target Amazon AWS, Microsoft Azure, Google AppEngine and Rackspace CloudServers. While anonymizing their findings, they analyze the collective of these cloud providers in terms of persistent storage (database interaction, price, etc.), elastic computing (cost per task, in terms of CPU, memory and disk usage, etc.), and networking (upload, download, cost, etc.)

2.3. Peering Agreements and Cloud Provider Interconnectivity

The aspect of internet peering and its implications is of increasing popularity in the research field and of great importance for a variety of reasons, from net neutrality to optimizing network traffic performance. While *transit* describes the act of one Autonomous System (AS) paying another AS, most often an internet service provider (ISP) to provide him with access to the public internet, *peering* on the other hand is the practice of two networks providers exchanging access to their networks and customers bi-laterally. The latter form of interconnection is the primary source of connecting two autonomous networks in the modern internet and is increasingly widespread [7]. While most forms of internet interconnection are still similar to those two peering options, the ever-growing nature of the market and technological advances have since brought a lot of different variations to those two practices into existence [8]. The job of physically connecting two networks to each other is usually done by an internet exchange point (IXP). Chatzis et al. [9] explain in their paper the importance of IXPs in regard to internet measurements as well as cloud and content delivery network performance. Giotsas et al. [10] present a detailed overview of the topic of IXP peering and its implications in regards to interconnecting the internet. They also present ways to determine IXP facilities and their geographical location using publicly available data and their CFS algorithm (Constrained Facility Search). The paper concludes, that this algorithm outperforms all other heuristics of pinning a peering activity to a physical facility with an accuracy of more than 90%. Arnold et al. [5] analyzed the aforementioned transit and peering interconnections on a vertical level and find, that the internets hierarchy is severely flattening, meaning that the traditional 3-tier model with Tier-1 ISPs as the backbone of the internet is

no longer the norm and is increasingly replaced by private interconnections. A big factor in those private interconnections are identified to be major cloud providers like Amazon, Google and IBM with their private infrastructure, which originally was intended to exchange traffic between their own datacenters, but is increasingly interconnecting with other providers and ISPs, ultimately even bypassing Tier-1 and Tier-2 ISPs. They cite, that those cloud providers are today capable of bypassing Tier-1 and Tier-2 ISPs entirely, while still reaching 76% of the internet. That brings up the question, whether this extensive private network benefits or hinders the reachability of these clouds from an end-user's perspective.

Arnold et al. [11] measure the performance differences between paths over the private WAN of two large cloud providers and paths over the public internet and find, that the vast majority (91%) have equal or improved performance using the private WAN option over the public internet, with 48% seeing an improvement. They also find, that those benefits generally improve with client-to-server distance and are very dependent on geographic location.

2.4. Internet Latency Measurements

To answer our research questions, multiple large-scale measurements must be taken from vantage points around the globe, to allow an analysis of latency and path structure between various home networks around the globe and the different datacenters of the cloud providers. There are several platforms, that provide a service allowing single users to do this.

RIPE Atlas The RIPE Atlas network has been used for measurements several times before. It and its functionality have been described and published by the RIPE NCC [12] and Bajpai et al. [13] have detailed the scientific value it merits and have described limitations to it. RIPE Atlas is a platform provided by the RIPE NCC, that lets private individuals as well as scientific researchers, corporations or others host small, physical probes and in return rewards them with the opportunity to make measurements in this network. A probe is a computer with a network connection, that can be used by the RIPE Atlas platform to schedule a wide array of measurements to one or more endpoints. Those measurements can include ICMP Ping measurements, ICMP or TCP traceroute measurements, HTTP/HTTPS measurements and others. Arnold et al. [11] have used it in their work alongside Speedchecker to determine the impact of private WANs on Cloud Performance. Giotsas et al. [10] have used it to map peering interconnections to facilities and have enhanced their measurement data with CAIDA IXP data and data from the PeeringDB. Gigis et al. [14] have used it with an exclusive focus on user connectivity and have shown, that RIPE Atlas is well suited to be used in measuring user-to-user latency, while Gedeon et al. [15] show, that it is useful to measure user-to-fog and - in extension - user-to-cloud latency, while also enriching their data with CAIDA.

Measurement Lab (M-Lab) Rajabiun et al. [16] use M-Lab to analyze the broadband infrastructure of Canada and Deng et al. [17] analyzed residential broadband capacity using

measurements from M-Lab. The latter also addresses problems that M-Lab has when measurements are taken by computers using NATs and can therefore isolate household measurements to accurately analyze their networking. M-Lab provides the ability to run tests from the user's machine to target points and makes the results publicly available [18]. It allows for Traceroute and Reverse Traceroute measurements, as well as their custom TCP Info measurement and their Network Diagnostic Tool [19]. Its knowledge base consists of all measurements taken from individual users and can be queried. As a successor to PlanetLab, it also incorporates their measurements [20]. Dovrolis et al. [21] describe it and its underlying software and presents it specifically as a tool for scientific network analysis.

Speedchecker A widely known and used commercial option for internet analytics measurements is Speedchecker [22]. It utilizes mobile hardware and software probes on PCs, Android phones and routers around the world and offers restricted access to schedule Ping, Traceroute, HTTP GET and DNS measurements. When signing up for their plan, one gets access to schedule those measurements from their selection of vantage points, spanning 170 countries and over 1000 ISPs. Chavula et al. [23] analyzed African intra-continental and inter-continental latency to speed test servers using Speedchecker. They found, that Africa could be clustered into distinct "latency clusters", in which latency is very similar across country borders. They also analyzed the peering set up by the local ISPs in those clusters and found, that this peering bears a significant performance impact on intra-continental latency, highlighting the importance of inter-country and inter-network connections in this area. Formoso et al. [24] contribute, that in many cases inter-continental connections are faster. A reason for this is identified as significant shortcomings in Africa's network design. They also use Speedchecker to make measurements and extract their information. As already mentioned, Arnold et al. [11] were using Speedchecker in combination with RIPE Atlas to obtain data monitoring the extent of the private WAN of cloud providers.

3. Methodology

In the following chapters, we are going to explain, what data we collected to answer the research questions. We will also detail, how we collected this data and go over the methods we used to clean and enrich the data and the frameworks utilized to do that.

3.1. Data Collection

3.1.1. Platform

The platform we chose for data collection was RIPE Atlas [25], as it is the *de-facto standard* for internet measurements in the research community right now. It provides the option of scheduling ICMP pings, as well as ICMP and TCP traceroutes. We included all these measurement methods in our data collection process. Atlas probes generally are installed in very heterogeneous network environments, e.g. home networks, research facilities and cloud datacenters. This allows us to observe the reachability of cloud providers on a global scale and among many different categories. By adding system and user tags to their probes, RIPE Atlas allows us, to filter out probes located within datacenters or focus just on home-based probes.

3.1.2. Vantage Points

Since we wanted our study to be as thorough as possible, we included all RIPE Atlas probes available as our vantage points. The distribution of RIPE Atlas probes across the globe,

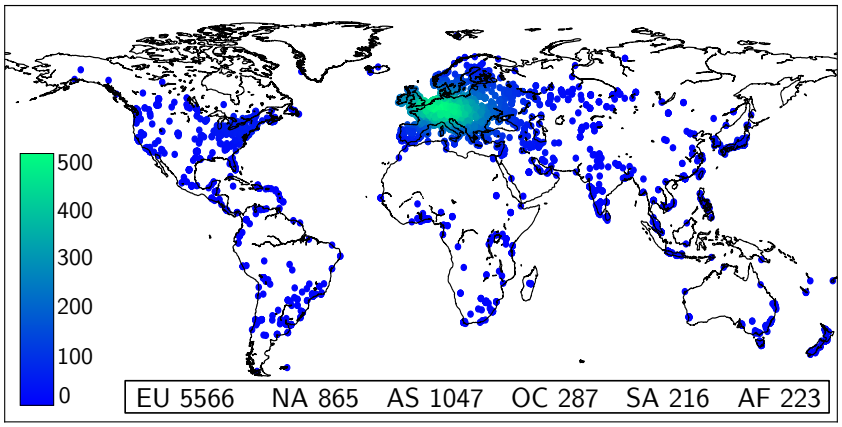


Figure 3.1.: Distribution of 8000+ RIPE Atlas probes used in our measurements.

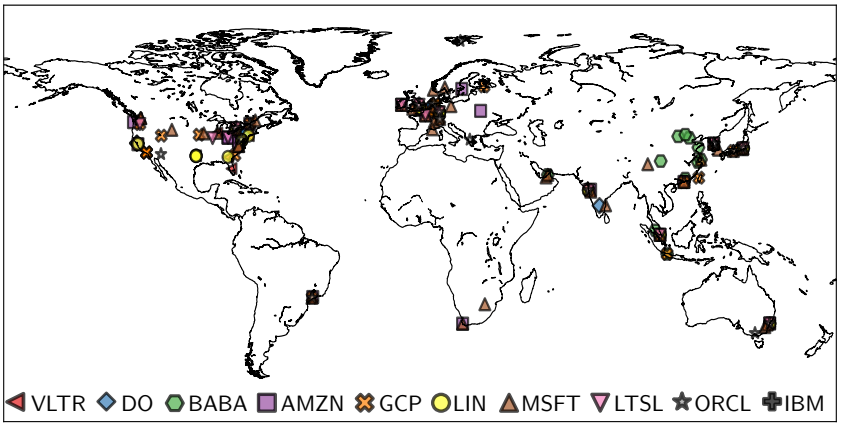


Figure 3.2.: Distribution of Datacenters by cloud providers (refer to table 3.1 for per-provider distribution).

including the number of probes per continent, can be seen in figure 3.1. As clearly visible, the overwhelming majority of probes are hosted in Europe. Continents like Africa, South America and Oceania on the other hand show a very limited density of probe distribution. This is a limitation of our measurements and will be addressed in chapter 5.3.

3.1.3. Cloud Providers & Datacenters

We list all chosen cloud providers and their deployments per continent in table 3.1. We selected every datacenter the chosen cloud providers had running by August 2020. The distribution of those datacenters around the globe can be seen in figure 3.2.

We chose ten cloud providers for our measurements. Our selection was focused on a broad variety of cloud providers with different focus regions, backbones and experience in the field. As they have been operational in the field for a long amount of time and also own resources, enabling them to utilize their own private backbone, we included Microsoft Azure,

	Datacenters per continent						Backbone
	EU	NA	SA	AS	AF	OC	N/W
Amazon EC2 (AMZN)	6	6	1	6	1	1	Private
Google (GCP)	6	10	1	8	-	1	Private
Microsoft (MSFT)	12	10	1	11	2	4	Private
Digital Ocean (DO)	4	6	-	1	-	-	Semi
Alibaba (BABA)	2	2	-	16	-	1	Semi
Vultr (VLTR)	4	9	-	1	-	1	Public
Linode (LIN)	2	5	-	3	-	1	Public
Amazon Lightsail (LTSL)	4	4	-	4	-	1	Private
Oracle (ORCL)	4	4	1	7	-	2	Private
IBM (IBM)	6	6	-	1	-	-	Semi
Total	**50**	**62**	**4**	**58**	**3**	**12**	

Table 3.1.: Datacenters per continent and provider

Google Cloud Compute Engine and Amazon EC2. As it is a younger offspring of the latter one, we also included Amazon Lightsail. As for smaller cloud providers utilizing the public internet as their backbone, we included the two best-established providers (to the best of our knowledge): Linode and Vultr. We also looked at other big tech companies making a more recent entry into the cloud computing field in the last years: IBM and Oracle. Lastly, we included Alibaba Cloud Computing, as we were interested in their specific focus on China and Asia in general, and Digital Ocean, a smaller provider, yet still employing their own backbone for a part of their path to the cloud.

When it came to finding an endpoint for our measurements, we had to make sure not to undergo load balancing, potentially rerouting our measurement to a different datacenter. For that purpose we used the vantage points of CloudHarmony [26]. It is a platform, that offers vantage points for internet measurements in virtual machines residing in the datacenters of all of the cloud providers mentioned above.

We scheduled measurements from every probe within a continent to every datacenter in this continent. We additionally wanted to inspect inter-continental connections for Africa and South America, so we scheduled measurements from probes in Africa to datacenters in South Europe and the USA and from probes in South America to the datacenters in the US.

3.2. Implementation

In order to schedule, parse, clean and enrich the data, we created the python package `ripeanalysis` (see chapter 6) for this thesis. In the following, we will describe its workflow (see 3.3).

To schedule measurements and receive the results outside of their web GUI, RIPE Atlas offers a RESTful API. Based on that API a python wrapper named cousteau exists. This library was used and improved by ERA [27], which was used to schedule measurements in this work. ERA allows the scheduling of multiple measurements from many probes to many target points. It also allows for a great amount of customization of the query, including native cousteau functionality. Most importantly it allows to schedule measurements from all the probes in a specific continent. ERA also allows the limitation of the number of probes per country when choosing all probes per continent. We have modified ERA to allow for TCP and paris traceroutes and to allow URLs as targets instead of IP addresses. That way we are able to write a script, which generates ERA commands based on our dataset of datacenters. We also limited the number of probes per country to 500.

We generated three ERA commands per continent - **one TCP traceroute, one ICMP traceroute and one ICMP Ping measurement** - for the continents North America, South America, Africa, Asia and Oceania and manually balanced them between the four API keys. For Europe, we needed to manually split the amount of probes into two different commands, since Europe exceeds 5000 probes and thereby the maximum amount of probes, RIPE Atlas allowed to use in one go. We also generated commands for the intercontinental measurements (Africa to South Europe, Africa to USA and South America to USA) and wrote all the commands into a

	Dates of measurements
1	September 17th 2020
2	September 23rd 2020
3	September 25th 2020
4	September 28th 2020
5	October 3rd 2020

Table 3.2.: Dates when measurements were taken.

bash script.

We scheduled the first set of measurements over a few days while still in the implementation phase, adjusting our scheduling routine after every continent until finishing the final bash file. This allowed us to schedule and import an entire measurement within a day. The dates on which we scheduled measurements are listed in table 3.2. Over those measurements, we collected 196 million datapoints from over 8500 vantage points to 189 datacenters.

The results were downloaded and imported using the standard RESTful web-API by a python script. The script queries RIPE Atlas for all measurements scheduled by the account, for which the API key is provided. The API returns a JSON file detailing measurement info for all measurements the account has ever scheduled. After analyzing the file, it filtered the returned measurement IDs by their timestamp. It compared them to the parameter "later_than", to only import the most recent measurements. Those measurement IDs are then divided by their type and individually downloaded from the RIPE Atlas API in their standard JSON format. (See full data structure in Appendix A.1 and A.2)

Afterward, those measurements were parsed, cleaned and enriched before being stored in an SQLite database.

3.3. Cleaning and Enriching the Data

3.3.1. Data parsing and Cleaning

To process the data, we first parsed it into a python data structure. We generated a list of traceroute objects, each having a list of hops, and a list of ping objects.

3.3.2. Parsing the Measurements

The data we parsed from the RIPE Atlas JSON responses is the following:

Traceroute measurements From the traceroute JSONs we parsed most importantly the destination name (**dst_name**), the destination address (**dest_addr**) and the source address (**from**) to identify the beginning and endpoint of the traceroute. In cases the **dst_name** field didn't return a URL, we overwrote it with the information in the target field of the previous

downloading step.

Further, we parsed information about the probe (**prb_id**) and measurement (**msm_id**), to later combine the traceroute with information about the vantage point and the measurement from RIPE Atlas, a **timestamp** and information about the protocol used (**proto**) as a way to distinguish ICMP traceroutes from TCP ones. Finally, we parsed the **paris_id**, which described the amount of different paris traceroutes RIPE Atlas allowed.

After gathering this meta-information, the hops were parsed. The information we selected was the hop number (**hop**), its source address (**from**), the TTL (**ttl**) and the RTT (**rtt**) for each hop. In case any hops come back erroneous, we discarded the traceroute. Any late hops were discarded as well. All unresponsive hops were marked as empty, the RTT and TTL values for this hop were set to 0 and the address was set to "x".

Ping measurements The most important part of the Ping measurement was again the destination name (**dst_name**), the destination address (**dest_addr**) and the source address (**from**), as well as the information about the probe (**prb_id**) and measurement (**msm_id**). Information about the protocol was not necessary, since only ICMP pings were possible via RIPE. We stored a **timestamp**, as well as the TTL (**ttl**) from the ping, although we later found out, that due to a bug in RIPE Atlas this information was unreliable.

For the actual pings, we imported the RTT (**rtt**). In case a ping came back erroneous, we defaulted the RTT to 0. In case the target timed out, we also defaulted to 0. The Ping data-structure we wrote in python (and subsequently the database layout, see chapter 3.4) supported up to five pings per measurement to provide scalability, even though for our usual RIPE Atlas measurements we decided to go with three pings per measurement.

3.3.3. Data Enriching

After parsing the traceroute and ping data into their respective data structures, we enriched the data with AS information. To do that, we stored all the IP addresses we encounter, including datacenter IP addresses, probe addresses, as well as every responsive hop address. We then filtered the addresses to exclude link-local addresses. Then the resulting set of IP addresses was enriched using the PeeringDB [1] and CAIDA dataset [28], as well as PyASN and their latest dataset [2]. Since the measurements were taken over a relatively short period of time, we downloaded the entire PeeringDB dataset and queried it locally. First, the .jsonl format was converted into a normal JSON array, before it could get parsed by python and converted into a dictionary. We parsed the Autonomous System Number (ASN) as a key and a tuple

[1] We extracted the PeeringDB dataset by querying the `peeringdb` python client for all ASNs. The dataset we used will be submitted with this thesis.

[2] The dataset was obtained by following the instructions on https://pypi.org/project/pyasn/, the version of the dataset we used is submitted with this thesis.

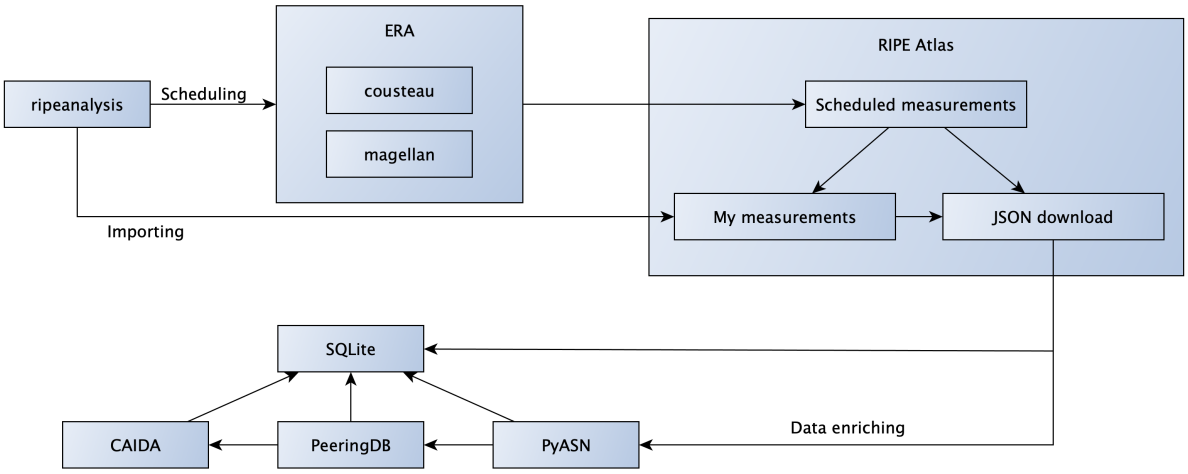


Figure 3.3.: Workflow of scheduling, importing and enriching the measurement data.

of the organization name and the organization type, that PeeringDB had for this ASN, as value.

If the IP address was known in the PyASN dataset and was thereby identified as part in an autonomous system (AS), we wanted to find out more details about this AS, specifically, whether it was part of an internet exchange provider (IXP). To get this information, we queried the CAIDA dataset, which we also downloaded for faster processing. We specifically queried the ixp_asns.json dataset and the ixs.json dataset. By looking up the ASN we previously got from PeeringDB, we then parsed further information, specifically whether the ASN belonged to an IXP, its CAIDA ix_id, its organization name according to CAIDA, as well as the region, country, city and coordinates of the organization's headquarter.

Besides this information, we also fetched the entire RIPE Atlas probe information from the RIPE Atlas API. This information contained, besides information about a probes location, architecture, firmware and connectivity, also several system-assigned and user-assigned tags, that helped to categorize these probes into groups. From this data, we imported the country a probe was sitting in and whether this probe included tags, which suggested it was either a home users probe (e.g. home, homelab, fritzbox), an organization-hosted probe (e.g. student, t-mobile, university), an IXP hosted probe (ix, ixp) or a datacenter hosted probe (e.g. datacenter, us-east1-b). A detailed list of these tags can be found in the Appendix (see A.1). We also looked up the continent associated with the country of the probe using the python `pycountry_convert` package. We filtered out all probes in the dataset, which were hosted in Antarctica.

Column name	Description
ID	Unique identifier; `primary key`
MSMID_ping	RIPE Atlas measurement ID for Ping
platform_used	Platform used; 'ripe' only in this dataset
protocol	Protocol used for ping measurements; 'ICMP' only in this dataset
timestamp	UNIX timestamp provided by RIPE Atlas, when the measurement was scheduled
url	Target URL
src	Source IP Address
dst	Destination IP Address
probe_id	Probe ID used by the platform to identify vantage point
ttl	Time-To-Live (TTL) after ping measurement
ping1 ping2 ping3 ping4 ping5	Ping measurements in [ms]

Table 3.3.: Database schema of `Ping` table

3.4. SQLite Database

The received data was stored in an SQLite database, since for such a big dataset, pre-filtering it using a query language like SQL before loading it into memory was necessary. The database structure was inspired by the way RIPE Atlas stored its measurements in a database [29], but modified to better suit our needs. The focus was also laid on expendability in case we wanted to extend this dataset to include more measurements or measurements from a different platform in the future.

The database split into three distinct parts: The ping measurements, the traceroute measurements and the meta-information. While the `ping` measurements were stored in one single table (`Ping`) containing all necessary information, the `traceroute` measurements had to be split up into three tables: The `Traceroute` table, the `TracerouteInfo` table and the `Hops` table. The meta information tables included the `NodeInfo` table, the `Datacenter` table and the `Probes` table.

Column name	Description
ID	Unique identifier; `primary key`
MSMID_Traceroute	RIPE Atlas measurement ID for Traceroute
platform_used	Platform used; 'ripe' only in this dataset
protocol	Protocol used for ping measurements; 'ICMP' or 'TCP' only in this dataset
timestamp	UNIX timestamp provided by RIPE Atlas, when the measurement was scheduled
src	Source IP Address
dst	Destination IP Address
paris_id	Paris ID provided by RIPE Atlas in case the measurement was scheduled with Paris Traceroutes; 0 otherwise

Table 3.4.: Database schema of `Traceroute` table

Column name	Description
ID	Unique identifier; `primary key`; `foreign key` to `Traceroute`
probe_id	Probe ID used by the platform to identify vantage point
cloud_provider	Name of the cloud provider, determined using ASN; `deprecated`
datacenter	ID of datacenter in `Datacenter` table, former `foreign key`, `deprecated`
url	URL of the target Datacenter, `foreign key` to `Datacenter`

Table 3.5.: Database schema of `TracerouteInfo` table

Column name	Description
ip	IP address of the node; `primary key`; `foreign key` to `Hops`
asn	AS number provided by pyasn for this IP address, *NULL* if none found
org_name_pdb	Organization name provided by PeeringDB for this ASN, *'Unknown ASN'* if none found
org_type_pdb	Organization type provided by PeeringDB for this ASN, *'Unknown'* if none found
is_ixp	Boolean value; 1 if the address belongs to an IXP, 0 otherwise
ix_id	CAIDA ix_id, used to refference following information, *""* if none found
org_name_caida	Organization name provided by CAIDA for this ASN, *""* if none found
region	Region of the facility provided by CAIDA for this ASN, *""* if none found
country	Country of the facility provided by CAIDA for this ASN, *""* if none found
city	City of the facility provided by CAIDA for this ASN, *""* if none found
latitude	Latitude of the facility provided by CAIDA for this ASN, *""* if none found
longitude	Longitude of the facility provided by CAIDA for this ASN, *"* if none found
facility_name	Facility name provided by CAIDA for this ASN, *""* if none found

Table 3.6.: Database schema of `NodeInfo` table

Column name	Description
ID	Unique identifier; `primary key`
name	Name of the cloud provider
url	URL of the datacenter provided by cloudharmony
country	ISO 3166 ALPHA-3 code of the country, the datacenter is located in
continent	ISO-alpha-2 code of the continent, the datacenter is located in

Table 3.7.: Database schema of `Datacenter` table

Column name	Description
hop_id	Unique identifier; `primary key`
src_ip	Source IP address of the hop.
dst_ip	Destination IP address of the hop
rtt_before	Round-Trip-Time (RTT) before hop, 0.0 before first hop
rtt_after	Round-Trip-Time (RTT) after hop
ttl	TTL (Time-To-Live) after hop
hop_number	Hop number to order hops of one Traceroute
attempt	Attempt number of hop (important to map hops to different Paris-Traceroute paths
Traceroute_ID	`foreign key` to `ID` in`Traceroute`

Table 3.8.: Database schema of `Hops` table

Column name	Description
ID	RIPE Atlas' unique identifier; `primary key`
country	ISO 3166 ALPHA-3 code of the country, the probe is located in
continent	ISO-alpha-2 code of the continent, the probe is located in
home organization datacenter ixp	Boolean; 1 if probe's user tags identifies with being of this certain category (see table A.1), 0 otherwise
longitude	Geographical longitude of probe
latitude	Geographical latitude of probe

Table 3.9.: Database schema of `Probes` table

3.5. Corrections

After finishing the measurements we noticed, that for unknown reasons Amazon Lightsail's cloud datacenters didn't respond to ICMP pings. Analysis of this cloud provider is therefore incomplete to some extend. The datacenters did respond to ICMP and TCP traceroutes, however, which is why we have listed them in sections, where we look at those measurements exclusively.

Table	Column
Ping	ping1 ping2 ping3 ping4 ping5 probe_id url
Datacenter	url
Hops	hop_number Traceroute_ID
Probes	probe_id
Traceroute	ID protocol
TracerouteInfo	ID url probe_id

Table 3.10.: Indices on tables in SQLite Database

Because of a classification error, measurements to the datacenter `me-south-1` (Google Compute Engine) were rendered unusable, so it was dropped from all analysis.

4. Results

In this chapter we will detail the results we found and attempt to use them to answer the research questions described earlier.

4.1. Global Cloud Reachability

4.1.1. Minimum Access to Cloud Network

We begin by analyzing the global reachability of the cloud and comparing the results to the latency categories detailed in chapter 2.1. Figure 4.1 plots the shortest latency of a probe in the region to the nearest datacenter, thereby constructing the best case scenario. We can see, that most parts of North America, Europe and Oceania can reach the cloud within 10 ms, which means they can reach the cloud within MTP latency. In addition, a huge part of Asia, including population rich countries like India, Russia and China also can reach the cloud within MTP. This correlates clearly with the huge investment from cloud providers into infrastructure in these areas. Especially China and India have seen an increase in local datacenter deployment compared to previous cloud measurements.

When analyzing Africa it becomes visible, that datacenter deployment on this continent heavily focuses on South Africa. Regions in the center and north-west of the country in contrast have the poorest latency connection in our measurements. Similarly to Africa, the datacenter deployment in South America entirely focuses on East Brazil, which is clearly represented in the latency in this region: Besides Brazil, which also achieves MTP latency,

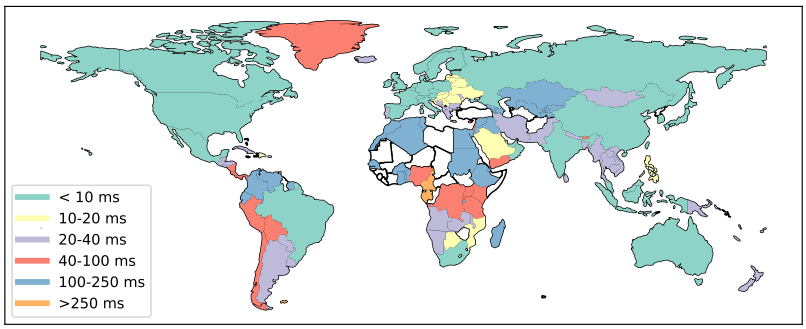


Figure 4.1.: Global minimum latency to nearest cloud datacenter

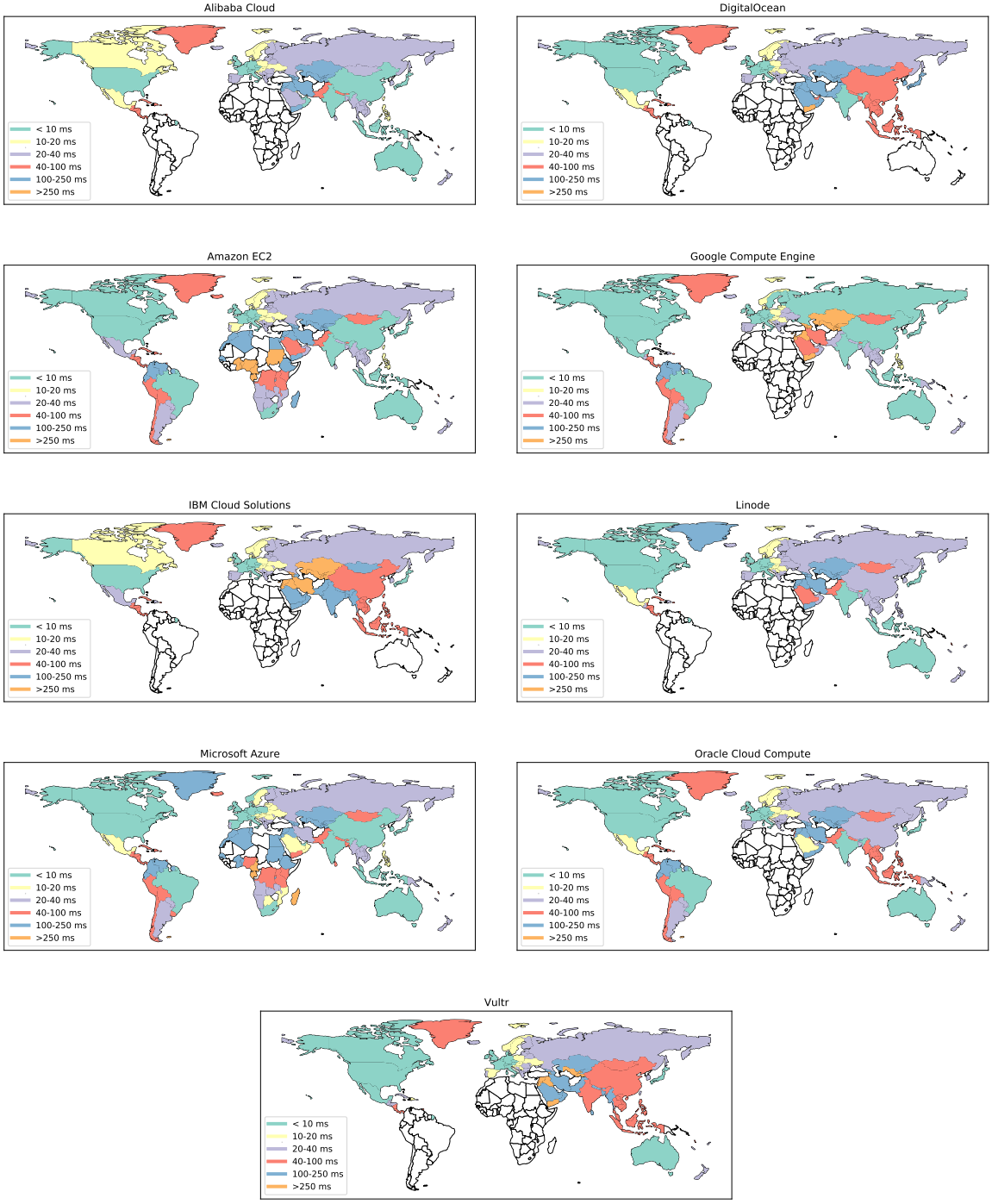


Figure 4.2.: Global minimum latency to nearest cloud datacenter per provider

all other countries have way lower best case latency. In contrast we can see that in Europe even countries without a locally deployed datacenter can reach the cloud significantly faster than those in Africa or South America. Reasons for this may include the geographically lower distance between the countries and the higher density of datacenters in Europe, but a stronger international network seems to factor in as well.

In general it can be observed, that a country with a best case latency lower than 10 ms has a datacenter deployed in it (compare with figure 3.2). This typically offers very low latency. Some countries even have more than one datacenter deployed in their region by the same provider (e.g. USA, Germany, Japan, China or Australia). To further investigate this correlation, as well as to compare different cloud providers in best case latancies, we plot the best case latency (as described earlier) per cloud provider in figure 4.2.

Right away we can see, that every single cloud provider is able to provide connections within MTP latency within the US and Central Europe. All cloud providers who are operating in Oceania can offer Australia the same service. The rest of North America also shows great best case latency. While some providers are able to provide MTP latency to the entire continent, all of them can provide minimum latency below 40 ms. One exception to this rule is Greenland, which seems not that well covered, though it still stays within perceivable latency with most providers.

Coverage of Asia varies hugely between providers. While most countries are covered within Human Reaction Time, countries in the Middle East seem to have the worst coverage, even falling below the 250 ms threshold. India on the other hand is covered fairly well and with the exception of Vultr shows MTP latency across providers. China's and Russia's best case latency on the other hand vary significantly depending on the provider. Interestingly, while only covered by half of the providers, South America shows similar coverage regardless of the provider, which seems intuitive given the similar deployment location. When it comes to Africa, the only intra-continental choices of coverage are Microsoft Azure and Amazon EC2. Here Microsoft continuously outperforms Amazon providing better best case latency in every single country but Madagaskar.

All in all, 45 countries can access the cloud within 10 ms and 21 countries can reach it in 10-21 ms. Combined with the fact, that those 66 countries hold a vast majority of the global population, this means, that most people can hypothetically connect to cloud computing services within MTP latency, which would open the possibility for applications like AR, VR and autonomous driving over the cloud [4]. Furthermore, 49 more countries are able to reach the cloud within 40 milliseconds and another 53 can reach it within 100 milliseconds, making all *but* 16 countries who host probes in our measurement able to connect to the cloud within human perceivable latency.

To see a more detailed distribution, we plotted the minimum RTT by all probes in our measurement set to the nearest datacenter grouped by continent as an ECDF in figure 4.3. The first interesting observation is, that around 45% of probes in Europe, North America, Asia and Oceania are able to access the cloud within MTP latency. It is also observable, that nearly all probes in North America and Europe and around 90% of probes in Oceania can reach the cloud within 50 ms, comfortably enabling applications requiring a delay within human

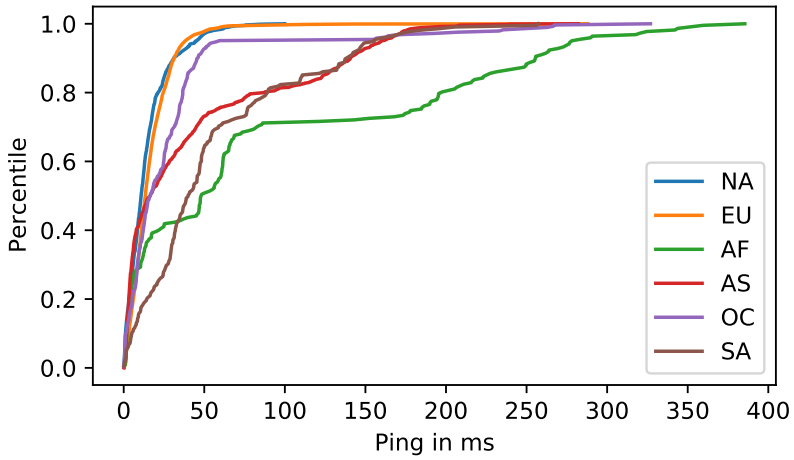


Figure 4.3.: Distribution of minimum RTT by all probes to the nearest datacenter grouped by continent.

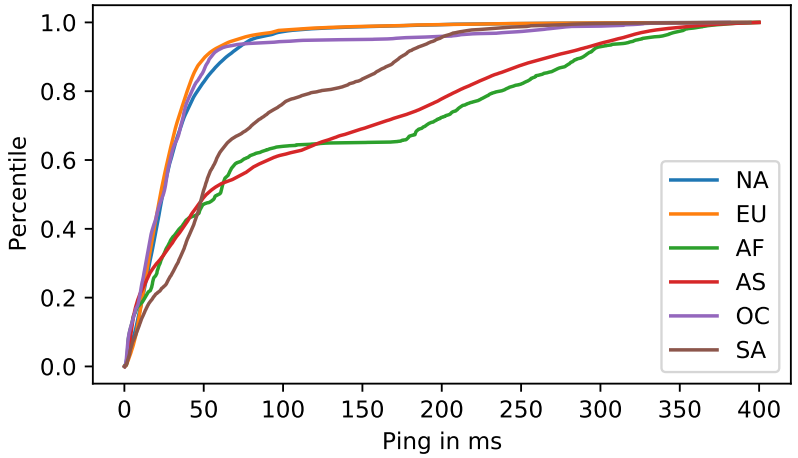


Figure 4.4.: Distribution of all RTT values recorded from all Atlas probes in our dataset to the closest datacenter.

perceivable latency. Interestingly, while starting out slower, 80% of Latin American and Asian probes can connect to the cloud within 100 ms. Only 70% of African probes can reach the HPL-threshold in the best case. Besides a few African probes though, all measurements were able to achieve latencies less than 250 ms. This result shines a good light on the cloud, even suggesting, that it is able to provide HRT applications to nearly every region in the world, including regions with sparse coverage such as Africa and South America.

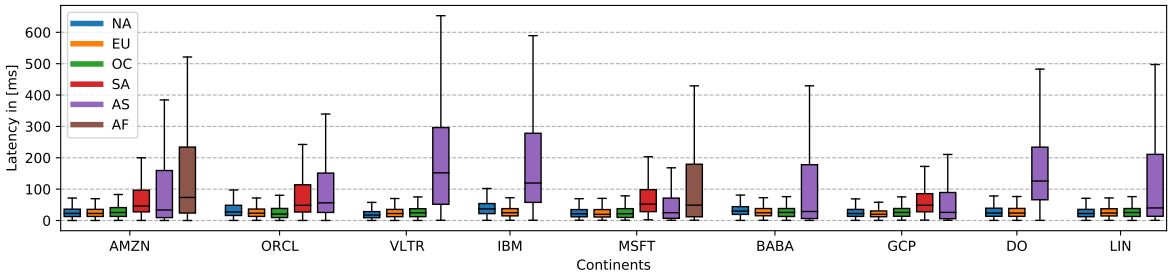


Figure 4.5.: Latencies to nearest datacenter per provider per continent

4.1.2. Overall Access to Cloud Network

Up until now we only looked at best cases. A slightly different picture is painted though when looking at the same ECDF, but now including all measurements to the closest datacenter, not just the minimal RTT. The resulting plot can be seen in figure 4.4.

Unsuprisingly, probes from Europe, North America and Oceania show rather excellent performance, satisfying MTP latency requirements in around 20% of measurements and accessing the cloud within 50 ms in around 80% of the cases. The differences are most notable for Asia. In reality only around 60% of measurements could reach the HPL threshold and around 15% couldn't connect to the cloud within 250 ms. The only continent performing worse is Africa with 20% of measurements failing to fulfill HRT requirements. When it comes to latencies of around 100 ms though, Africa even outperforms Asia narrowly with around 62% reaching the cloud within HPL.

Those results are roughly in line with the results from our best case analysis and can be explained by the density of the cloud datacenter deployment in regions like North America, Europe and Oceania in contrast to the lack of adequate coverage in Africa and South America, as well as the worse network interconnectivity between countries in those respective regions. The widely varying performance in Asia can be tracked back to some countries such as Singapore, South Korea and Japan having quite good coverage, while other countries are wildly less covered. Those countries - predominantly in the middle east - have weaker network infrastructure and are geographically far away from their next datacenter.

To compare those results by providers, we plot their performance in every region in figure 4.5. For Europe, North America and Oceania we can see, that results up until the 95th percentile consistently come in below the 100 ms threshold for every provider without major variations between them. This is mainly due to the cloud providers' extensive deployment in those regions and the public network infrastructure in those regions. This leads to differences between private and public backbone providers being indistinguishable. Within Africa, Microsoft Azure shows slightly better performance due to the fact, that they employ two datacenters in comparison to Amazon's single datacenter. For South America results look similar across the board with similar median and quartiles for all providers. Looking at Asia, though, the real-life performance varies quite significantly between providers. While some bigger providers like Microsoft and Google achieve latencies with 95th percentile around

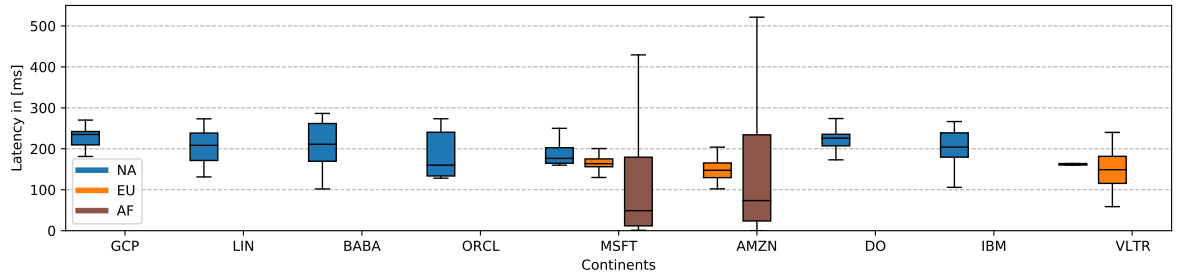


Figure 4.6.: Minimum cloud access latencies from probes in Africa to closest datacenter in the US and Southern Europe (compared to intra-continental measurements)

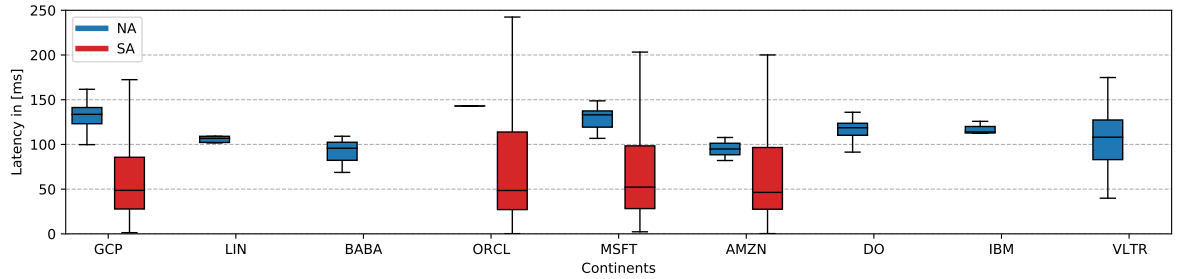


Figure 4.7.: Minimum cloud access latencies from probes in South America to closest datacenter in the US (compared to intra-continental measurements)

200 ms, smaller providers like Digital Ocean and Linode can reach results up to 400 ms and Vultr and IBM both reach around 600 ms on the 95th percentile. This variance can again largely be explained by the drastically lower datacenter deployment in Asia compared to landmass and area and the comparatively wide spread of population.

4.1.3. Inter-Continental Access to Cloud Network

Lastly, we take a closer look at the measurements we collected from Africa to South Europe and the US (figure 4.6), as well as from South America to the US (figure 4.7). Those continents both lack international network interconnection. Especially in Africa, large parts of the continent are disconnected. Even South Africa, which sees 100% of the continents datacenter deployment, has limited coverage. On the other hand, inter-continental connectivity has developed rapidly over the last years, with extensive transatlantic connectivity at the forefront. With these measurements we aim to find out, whether a strong transatlantic backbone can outperform the physical proximity to the datacenter.

Looking at Africa first, we can see clearly, that the huge geographical distance, which network traffic must travel from Africa to Europe, reflects in latencies that are overwhelmingly often beyond 100 ms, in nearly all cases over 50 ms. This clearly doesn't seems to be a viable alternative to intra-continental cloud providers, which by comparison could achieve lower

latencies for 60% of the probes. But the fact, hat Africa is a geographically large continent and the datacenter deployment heavily focuses on the south of the continent, lets us assume, that those lowest inter-continental latencies are measured from probes in Northern and Central Africa. Those probes probably had much higher latencies to their intra-continental datacenters, compared to its inter-continental connections. In that regard we can conclude, that - while maybe an option for regions in Northern and Central Africa - inter-continental connectivity in Africa is outperformed by its intra-continental datacenters for an overwhelming part of users.

For South America, while the spread of results is much more tightly focused around latencies between 50 and 150 ms depending on the provider, it is also not a fundamental improvement to latencies achieved with datacenters in Latin America. In that regard it also confirms our expectations, since the datacenters in South America are heavily focused around the country of Brazil, which offers a geographically shorter distance to any probe in Latin America, than any US datacenter can do, which is reflected in the resulting higher latency.

4.2. Impact of Cloud Deployment Density

To further compare the differences in cloud accessability between areas with dense deployment of datacenters and areas with sparser datacenter coverage, we open two case studies:

4.2.1. Case Study A: The United States of America

We first look at the United States of America, since they have the most datacenters deployed in any country. Specifically, we look at the Primary Statistical Areas in the U.S., cities and areas, which contain a large amount of people [30, 31]. We chose the top 98 PSAs, which collectively house more than 80% of the population. We then selected all RIPE Atlas probes within a 150 km radius of these PSAs and selected all ICMP ping data from those probes to their closest datacenter. We plot the minimum latency, the median and the 95th percentile of our results in figure 4.8.

The minimum distribution shows virtually all probes reaching the cloud within 25 ms, with 50% coming in under 10 ms, surpassing our MTP threshold. Furthermore, the median distribution shows nearly all probes accessing the cloud within 50 ms, comfortably enabling HPL-requiring applications. Even the 95th percentile distribution, which should include measurements taken under imperfect network conditions, could achieve perceivable latency requirements in more than 65% of the case and was able to provide access latencies within human reaction time for more than 95% of the probes measured.

This case study shows us, that a majority of the US population lives in areas, where the cloud can be accessed within Human Perceivable Latency, making it ready for a bulk of next-generation cloud applications. Furthermore, a huge portion of the population can already reach the cloud within Motion-to-Photon latency, fulfilling the requirements for applications like autonomous driving.

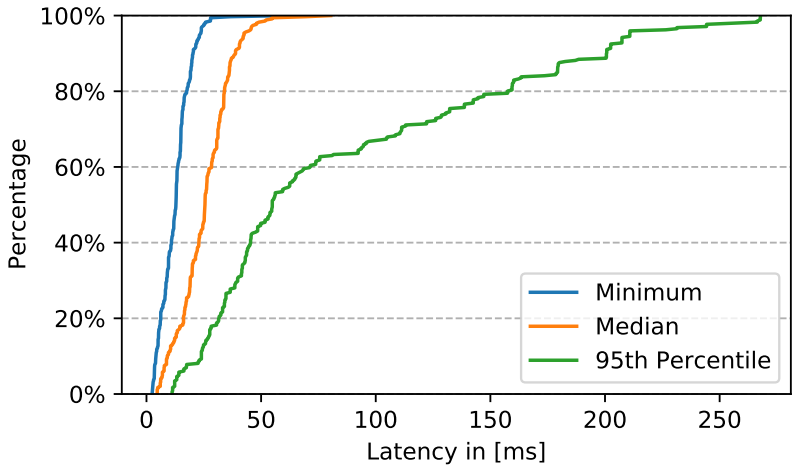


Figure 4.8.: Minimum, median and 95th percentile distribution of measurements within a PSA

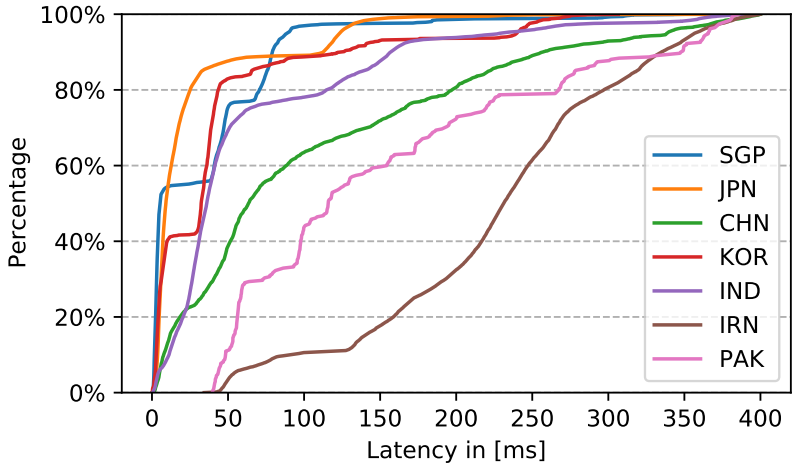
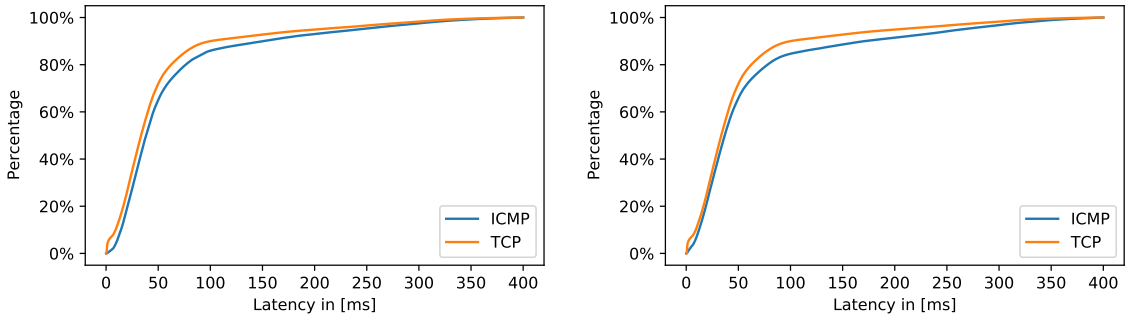


Figure 4.9.: Distribution of latency in a subset of Asian countries with and without in-land datacenters

4.2.2. Case Study B: Asia

We now compare our findings from case study A with measurements taken from certain regions in Asia. We look at some countries, that have multiple datacenters from different providers deployed in their country (China, Japan, South Korea, India and Singapore), as well as Pakistan, which only shares borders with a country, that has a datacenter deployed (India). We also include Iran, since it is the country farthest away from any datacenter in the region. We plot the pings achieved by probes in those countries to their nearest datacenter as a distribution in figure 4.9 .



a) ICMP vs TCP pings

b) ICMP vs TCP traceroutes (last-hop RTT)

Figure 4.10.: Latency comparison between ICMP and TCP

We can immediatley make some key observations: Firstly, the fact that smaller countries (South Korea, Japan and Singapor) have their datacenter deployments in very densly populated areas is clearly visible from the fact, that between 40 and 60% of the measurements in this area reach the cloud within MTP latency and around 90% can access it within human perceivable latency. Bigger countries on the opposite have a far more widely distributed population, making the latency distribution curve way less steep. Still, a majority of the people (60-80% of probes) can reach the cloud within HPL constraints. Pakistan on the other hand has a way lower latency distribution, enabling perceivable latency applications in only 35% of measurements. Further it has to be noted, that no measurement was faster than 30 ms, making MTP applications impossible. Finally, being the farthest away from the nearest datacenter, Iran's probes could only 10% of the time access the cloud within 100 ms. 40% of the samples couldn't even satisfy HRT constraints.

We conclude, that areas with dense datacenter deployment can consistently expect smaller cloud access latencies. We further conclude our assumption, that a shorter geographical distance to the nearest datacenter improves latency significantly.

4.3. ICMP vs. TCP Measurements

The scientific consensus at the moment is, that ICMP measurements are more optimistic then TCP measurements, since ICMP responses usually don't undergo any form of throttling to prevent congestion. To analyze this, we compare our ICMP and TCP measurements. As described in chapter 3, we scheduled ICMP ping, ICMP traceroute and TCP traceroute measurements. The latter we also use, to get a fairly accurate TCP ping measurement by taking the last-hop RTT. When talking about TCP pings, we are talking about this last-hop RTT.

We first plot our TCP and ICMP ping measurement results, both times excluding responses that came within 0 ms, as this is our default value for erroneous measurements. The results

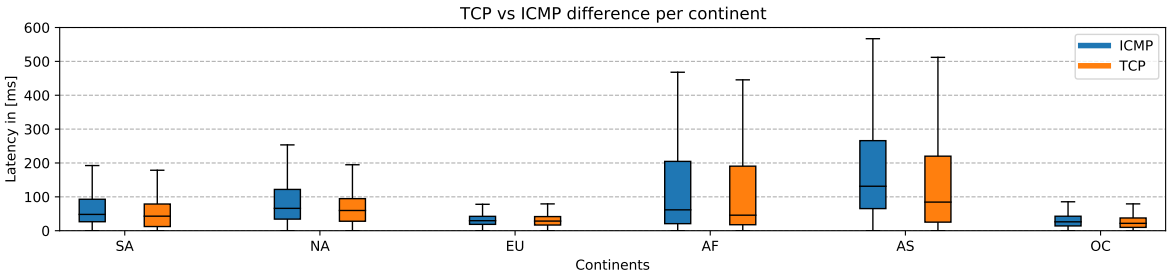


Figure 4.11.: Global latency comparison between ICMP and TCP per continent

can be seen in figure 4.10a. As evident, while following the same general distribution, TCP measurements are consistently lower then their ICMP counterpart. The trend can also be reproduced on a per continent basis, further solidifying the finding (figure 4.11).

To make sure, that the way we are measuring TCP pings isn't biased, we also plot last-hop RTTs of both ICMP and TCP traceroutes (figure 4.10b). The results offer a similar distribution. If you take into account, that some providers like Amazon Lightsail outright don't respond to ICMP pings all together, this suggests, that cloud providers have taken a new approach to handling ICMP traffic differently then TCP traffic. Since HTTP traffic utilizes TCP as underlying protocol, we believe, that our latency measurements from TCP traceroutes are more reflective of user cloud access times.

4.4. Differences in Underlying Infrastructure

In the last decade, big cloud providers have invested billions of dollars into expanding their private networks. The intention behind this is to route the traffic of their customers around the public internet. Those shielded networks serve the purpose of avoiding congestion, improving transmission rates and optimizing their user's path to the cloud. In this section, we analyze the impact of those extensive network deployment efforts and investigate, what percentage of the path from user to cloud is already controlled by the cloud providers. We have already classified the cloud provider's backbones in table 3.2.

To find out, which hops belong to a cloud provider's network, we look up the ASN of each hops' IP address and its respective owner. A list of PeeringDB organization names associated with the cloud providers can be found in the appendix (table A.2).

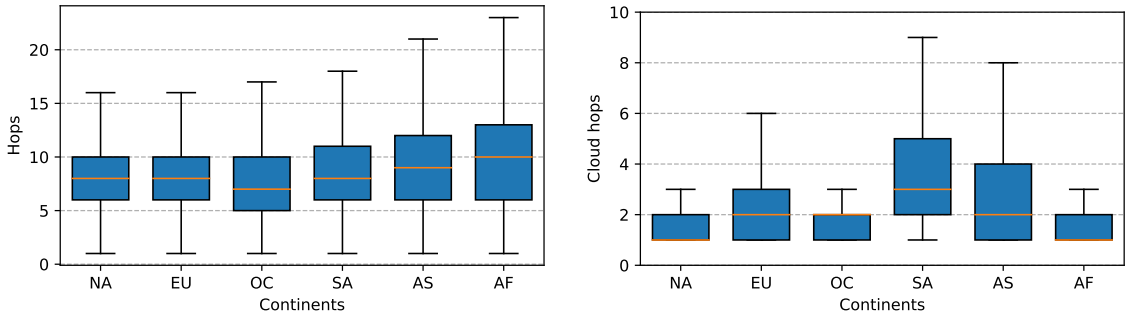
Next, we plot the total path length (meaning all hops excluding unresponsive hops and link-local addresses) in comparison to hops, which could be identified as belonging to a cloud provider. The results can be seen in figure 4.12. Right away it is clearly visible, that continents with lower latency have shorter paths to the cloud. In particular North America, Europe and Oceania show a tighter path length distribution than Asia and Africa, with South America in the middle, tending towards shorter path lengths.

When looking at the hops belonging to cloud providers, the results are way more surprising. While Africa sees a very low amount of cloud-hops in its traceroutes, which suggests the

cloud providers' WANs (all datacenters in the region are employed by Microsoft and Amazon, both using private WANs) aren't as developed in this region and is very much expected, other more widely covered regions like Asia and Europe have a way longer distance traveled in the cloud providers network. This in turn suggests high investment in the private WAN in those areas and, again, is very much expected.

More unexpected though is the relatively short path length within the cloud providers' networks in the regions North America and Oceania, as well as the outstandingly high one in South America. A possible reason for this might be, that many providers, both with and without private backbone - are hosting datacenters in North America and Oceania. Consequently, the providers without private backbone are pulling down the average path length within the providers network. On the other hand, only bigger providers - utilizing a private backbone - are operating in South America, raising the average.

To look into this more accurately, we want to compare the pervasiveness of the cloud per



a) Path length per continent

b) Hops belonging to cloud providers by continent

Figure 4.12.: Path length and hops belonging to cloud provider

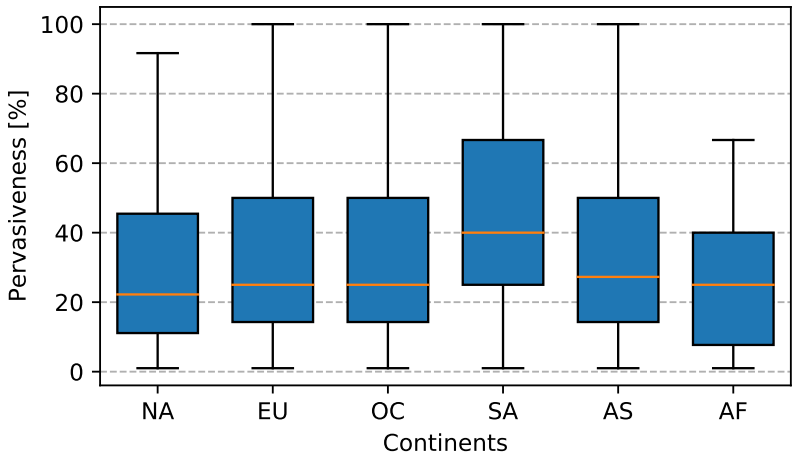


Figure 4.13.: Pervasiveness of path to cloud per continent

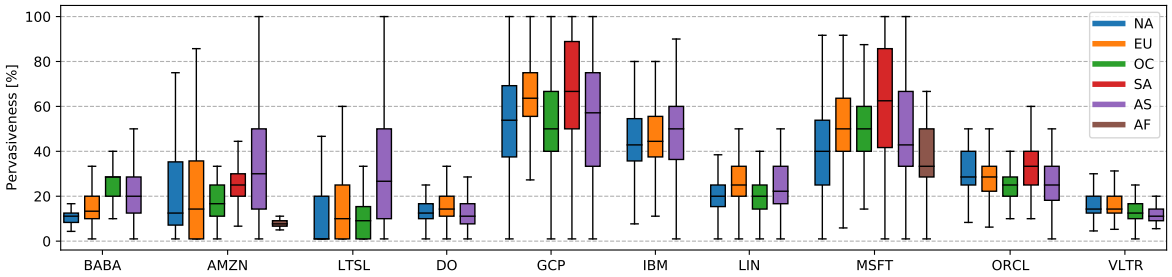
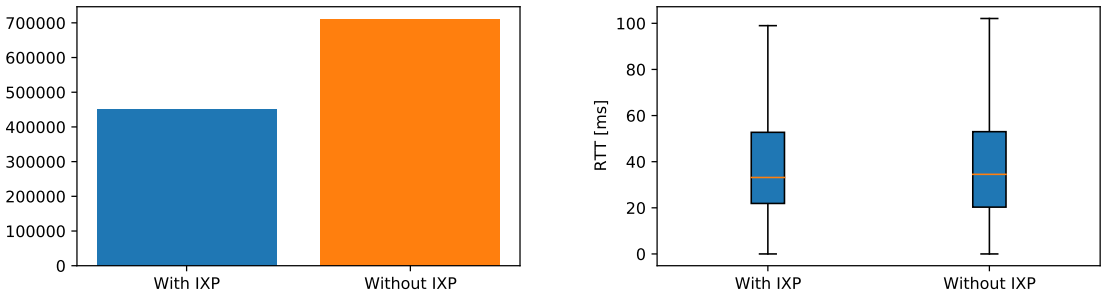


Figure 4.14.: Pervasiveness of path to cloud by providers and continents

provider per continent. We define pervasiveness as the the amount of hops made within a cloud providers' network divided by the total amount of hops traffic has to go through to reach a certain datacenter. The pervasiveness of the cloud per continent is plotted in figure 4.13. It can be observed, that the cloud providers already own on average 20 - 40% of the path to the cloud. It is also very common for clouds to own more than 50% of the path, even reaching up to 100% in some cases, meaning the first public IP address encountered in the path belongs to a cloud provider. This might occur, when a probe is located within a datacenter of the same cloud provider, but the fact this happens in at least 5% of the cases suggests that this is also true for some outside probes.

Comparing pervasiveness between continents, most performances look similar. Outstanding results are Africa, which shows way lower pervasiveness, probably due to the low maturity of the cloud providers networks in the region, South America, which has a higher than average median pervasiveness, and North America with a lower than average pervasiveness. We assume, that both outliers are rooted in the type of backbone employed by the cloud providers in the region: North America, which hosts datacenters from all cloud providers in our study, including ones, which use the public internet as backbone, pulling down the average. South America on the other hand only hosts datacenters from private-WAN employing providers.

To further investigate, we plot the pervasiveness of the path to the cloud for all providers on all continents in figure 4.14. Here the differences between private and public WAN are clearly visible. For providers utilizing the public internet as a backbone, e.g. Linode and Vultr, the pervasiveness tops out at 55%. The same applies for a majority of the providers classified as 'Semi-Private Backbone', utilizing both private backbones in some areas and public backbones in others. The exception to this rule is IBM, showing up to 80% percent pervasiveness, which suggests, that their private WAN is already very extensive. Cloud providers utilizing private backbones to their fullest extend, e.g. Amazon (Lightsail and EC2), Google, Microsoft and Oracle on the other hand exhibit a high degree of pervasiveness, even reaching up to 100% pervasiveness. This is differs between continents for some providers.



a) Measurements with and without IXP hops

b) Latency comparison

Figure 4.15.: Traceroute measurements with and without IXPs

4.5. Impact of ISP Peering Agreements on User Cloud Access Latency

To make their aforementioned private backbone accessible to the user, cloud providers have to connect their network to the user's Internet Service Provider (ISP). To forego public IXPs and to get their network as close to the customer as possible, cloud providers set up peering agreements with those ISPs directly. In this section we want to analyze the impact of those peering agreements and compare them to network traffic routed through IXPs.

To analyze the impact of ISP peering agreements from a user perspective, we only look at home-based probes for our analysis. We analyze their traceroutes and identify all hops owned by an IXP. In this context we found, that most cloud providers are already classified as IXPs by the CAIDA dataset. In the following we consequently only classify a node as IXP, when it is both classified by CAIDA as an IXP and doesn't have the organization name of a cloud provider (see list A.2). We assume, that user traffic travels from their ISP onwards straight to the nearest cloud network access point. The "traditional" way of doing so would be to go through an IXP. But with the rise of the "flat internet", more and more cloud providers are peering directly with the ISPs. Therefore, if there is no IXP node within the traceroute, we assume a peering agreement in place.

We filter our dataset as described. As a result, we get around 1.2 million traceroute measurements. Figure 4.15a shows, that around one third of the traceroute measurements contains at least one IXP node, indicating an ISP peering agreement for around one third of traceroutes. We also take a look at the latency differences in figure 4.15b. While looking similar, latency of paths, that go through an IXP node see a slight advantage in terms of lower median latency.

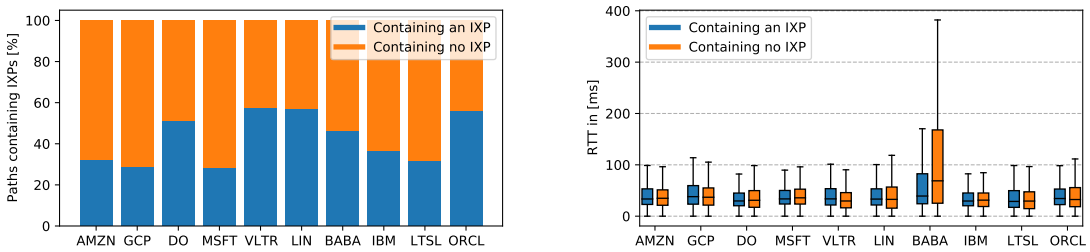
To analyze this more closely, we look at the differences in IXP peering between providers. Here, the results show a clear difference between some providers. Figure 4.16a shows the percentage of paths containing IXP nodes. While Amazon (EC2 and Lightsail), Google and Microsoft Azure utilize IXP peering only around 30% of the time, Linode, Vultr and Oracle

need to rely on them in 60% of our measurements. When we compare these findings with table 3.1, a clear trend is visible: Cloud providers with private backbones have a way lower usage of IXP nodes, than providers with a backbone relying on the public internet do. Those providers utilizing a mix between those two, can be found right in between (Digital Ocean, IBM, Alibaba). This is in line with the philosophy of the bigger cloud providers, to segregate their traffic away from the public internet as much as possible. It also explains the high pervasiveness, that those cloud providers showed in chapter 4.4.

We now want to focus on the impact those peering agreements have on the user to cloud connectivity. The first one is bandwidth: A direct peering between ISP and cloud provider allows for provisioning huge bandwidth links to cloud traffic. Furthermore, those links are also isolated, giving the cloud provider the full control of managing his traffic with this bandwidth free from other competing traffic. Since none of our measurements were aimed to measure bandwidth, we cannot prove this theory.

The second factor cloud providers hope to optimize with peering agreements is latency. The idea is, that less congestion at high traffic IXPs and more end-to-end control over the path automatically improves the user-to-cloud latency. Figure 4.16b compares latencies between paths via an IXP and paths with peering agreements in place. Interestingly, we can see, that peering agreements have virtually no effect on latency. Especially for cloud providers with a private backbone, the differences are hardly noticeable. Small differences can be observed for providers using the public internet as backbone, though the effect varies wildly and most often benefits the median mildly, while simultaneously having a bigger variance. One notable exception in that context is Alibaba, for which the peering agreement is not beneficial at all, showing higher latencies from the median upwards. This is probably due to the fact, that Alibaba operates its private WAN only in parts of Asia (mostly China) and employs peering agreements in that region, while in Europe and the US it uses the public internet. The peering agreement measurements for Alibaba consequently originated in Asia, where RIPE Atlas probe coverage is sparse, resulting in longer paths.

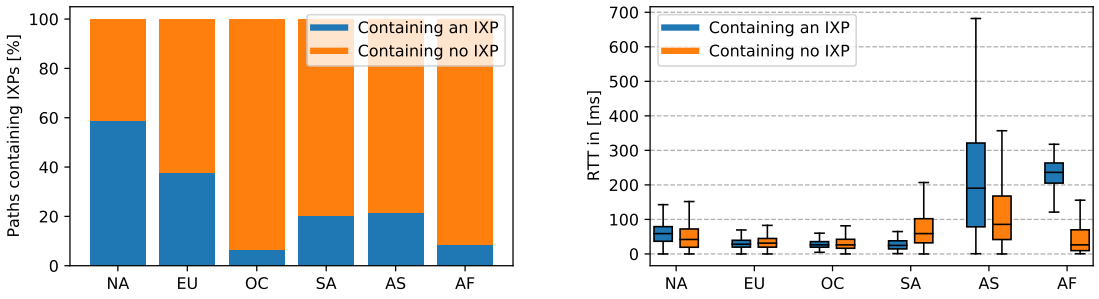
Lastly, we want to compare the impact of ISP peering agreements on latency per continent. Figure 4.17a shows the percentage of peering agreements per continent. Similar to our pervasiveness study, we can see, that regions, which are considered focal points and



a) Measurements with and without IXP hops

b) Latency comparison

Figure 4.16.: Differences in IXP peering per provider



a) Measurements with and without IXP hops

b) Latency comparison

Figure 4.17.: Differences in IXP peering per continent

thereby host datacenters from many providers (NA and EU), including those without private backbone, have a higher rate of IXP-based routing as others. On the opposite side, regions with predominantly private backbone cloud providers have peering agreements in place between 80 and 90% of the time. As we can see in figure 4.17b, this can have some impact on latency. In regions, that don't have a good and highly interconnected public internet infrastructure in place (Africa, Asia), peering agreements can indeed greatly benefit latencies, while in regions with better public internet connectivity (EU, NA, OC), peering agreements show limited usefulness.

To confirm this theory, we look at the latencies per continent for some providers, specifically those with a private backbone. We chose Microsoft, Amazon (EC2 and Lightsail) and Google, because they deploy datacenters in the most continents and have a similar pesrcentage of peering agreements within their traceroutes. The results are ploted in figure 4.18. We can see, that the results for these providers are similar to the overall-picture we got in figure 4.16b. For the continents with well-developed networks, peering agreements only have minor influences on latency. In parallel, we see that those continents are the least interesting for the cloud providers to make peering agreements in. For Africa and Asia on the other hand we see clear improvements achieved by using peering agreements. We can also see, that in those continents, a larger portion of traffic comes in via peering agreements.

Two exceptions to this are Oceania and South America. Oceania is known for a quite good internet connectivity, nevertheless peering agreements are responsible for nearly all network traffic to cloud providers in this region. Similarly to North America and Europe though, the latency differences are minute; the visible differences are mostly due to a lack of measurements with IXP nodes. In the case of South America, we see, that peering agreements seem to have a negative impact on latencies. This might again stem from a relatively low amount of measurements with IXP nodes within the paths and from the relatively bad coverage of probes and datacenters in South America, slimming down our measurements. This result might get clearer, once more measurements are taken.

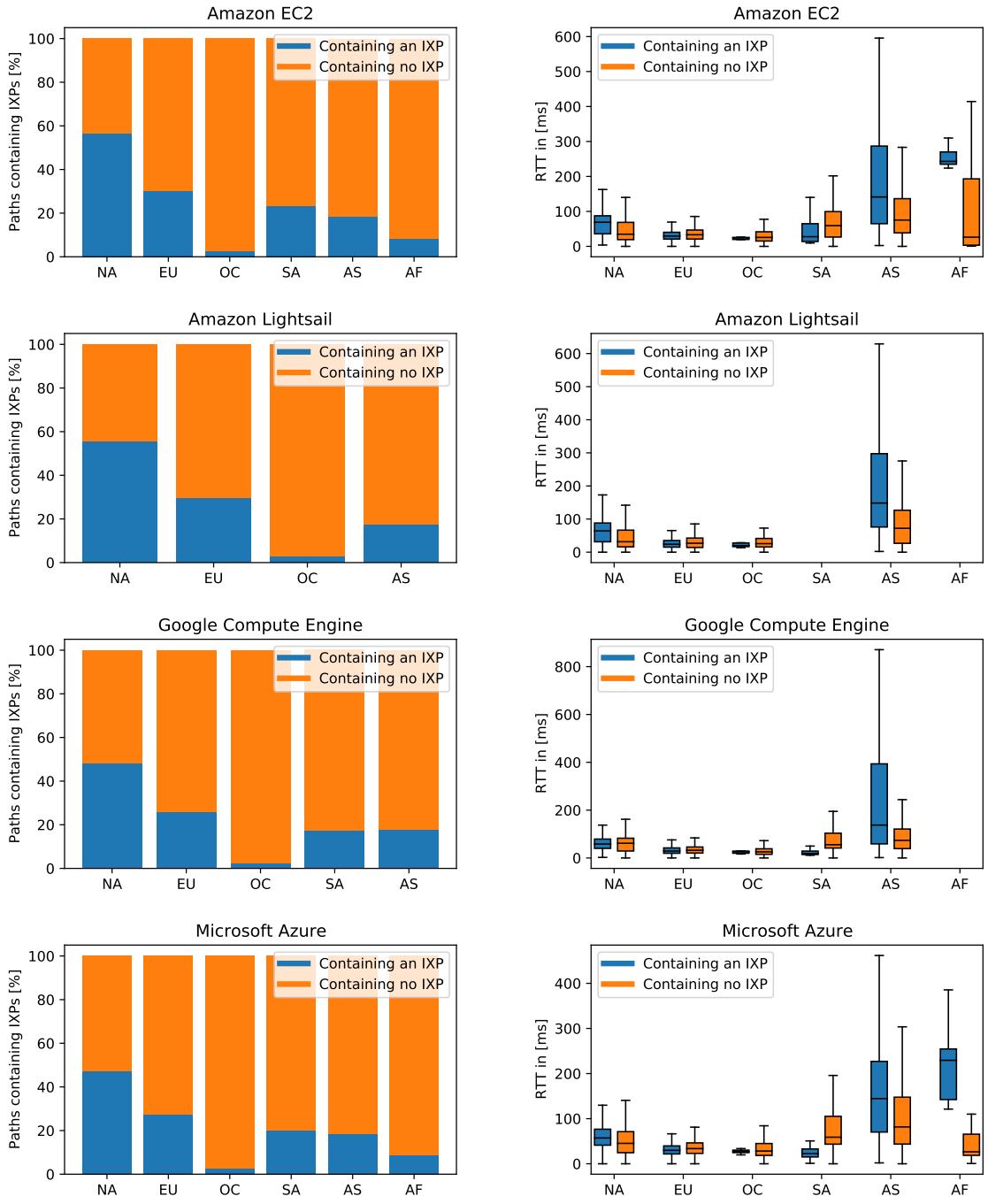


Figure 4.18.: Differences in IXP peering per continent for selected providers

5. Conclusion & Discussion

5.1. Conclusion

RQ1: What are the user access latencies to the current cloud infrastructure across the globe? We have observed a great variance in the user-to-cloud access latency between continents. Chapter 4.1 has shown, that Europe, North America and Oceania do show great performance, often even achieving latency low enough to enable applications like AR and VR directly from the cloud. The infrastructure in continents like Africa, South America and Asia on the other hand clearly still has to mature. South America and Asia show promising results, often providing access latency below the threshold of human perceivable latency. However, Africa's performance is still held back by its poor international network infrastructure and the sparse deployment of datacenters in the region. Nevertheless, around 80% of probes could at least satisfy the constraints of human reaction time.

We conducted a case study of the primary statistical areas in the US and found, that around 50% of the measurements in this area had the potential to reach the datacenter within Motion-to-Photon latency, while even at the 95th percentile 65% of probes were able to achieve latencies below 100 ms. We also conducted a case study of a number of Asian countries with different cloud deployment and population characteristics and concluded, that geographical distance to the cloud is a major factor of user to cloud access latency. We can also say, that for this reason datacenters within the continent usually outperform those in other continents.

RQ2: Does the type of underlying networking backbone interconnecting cloud infrastructure impact user cloud access? We analyzed the difference in latency performance between providers, specifically focusing on the difference in underlying backbone in chapter 4.1.2. We did find, that providers, who do employ a private backbone have comparatively lower latencies in regions with poor network infrastructure. However, they don't show significantly lower latencies in areas like North America, Europe or Oceania, where the public internet's infrastructure is fairly good.

We can also conclude, that cloud providers using private WANs own increasing parts of the path to the cloud. Chapter 4.4 has shown, that cloud providers can even reach up to 100% path coverage in some cases. This effect varies from continent to continent and between providers.

RQ3: What is the impact of ISP peering agreements on global user-cloud access? We looked at traceroutes, that we could identify as going through an IXP node, and compared them with paths, where we identified a peering agreement between cloud provider and ISP

in place. Chapter 4.5 showed, that around two thirds of all cloud traffic is routed through such peering paths and have shown, that the amount of peering agreement paths is higher for providers with a private backbone in place. We have found, that peering agreements have virtually no impact on latencies in regions with good network connectivity, but improve latencies in regions, that don't.

5.2. Limitations

The limited nature of this thesis only allowed for a total set of five measurements, which - while already providing enough datapoints to eliminate outliers in general - could be improved by running more measurements over a greater period of time. In combination, one has to consider the enormous size of the dataset and the consequences of growing it any larger in terms of computing and storage capacity .

We also have confined ourselves to the RIPE Atlas measurement platform. The tools we created and the underlying database structure are designed to be open to the contribution of any other measurement platform, a few of which we have already mentioned in chapter 2. The comparison of those measurements and the RIPE Atlas measurements made in this Thesis project could eliminate any bias on the way the measurements were taken. In this context, we mentioned the uneven distribution of RIPE Atlas probes across the globe. Further measurements from other platforms will eradicate this bias.

Our research on ISP peering agreements was entirely based on what we were able to infer from our traceroute measurements. Since we don't cross-reference our findings with other public datasets, our investigation on this topic is quite crude.

As detailed in chapter 3.5, the project unfortunately had to exclude some providers and datacenters from this study. A further analysis into those could provide further insights.

5.3. Future Work

As previously mentioned, the open structure of the project results enable a vast array of possibilities to continue this research topic. This includes the expansion of the database by adding more and frequent RIPE Atlas measurements, as well as the addition of new measurement platforms to the dataset. Both would bring inherent value and would increase confidence in the resulting findings. A long term continuation of the measurements might even show the increased investments cloud providers take over the next few years and evaluate their usefulness. We also think that the dataset as it is can be used for further analysis in the workings and developement of cloud providers over the years.

6. Reproducibility

6.1. Environment

Research was conducted using Python 3.8, Jupyter 4.6.3. Jupyter Notebook 6.0.3 and SQLite 3.33.0. All installed Python packages are listed in table 6.2. The ERA package is not yet publicly available. All scripts required to generate ERA commands and import the RIPE Atlas measurements into the SQLite database reside within the `ripeanalysis` Python package. All Jupyter Notebooks used to generate the plots for the thesis are in the same package. A directory tree can be seen in figure 6.1.

6.2. Reproduction

To schedule the measurements via ERA, commands were generated using the `ripeanalysis/era/era-gen.py` script. The commands used by us are commented below. To import the measurements into the SQLite database used for analysis, run the `ripeanalysis/database/import_ripe_measurements.py` script. The imported data can be found in `ripeanalysis/database/SQLite/ripeanalysis-data.db`. Valid API keys for RIPE Atlas must be added in every script. In the import script, a timestamp must be given, after which measurements should be imported. If all measurements should be imported, set this to 0.

To generate the plots, run the corresponding Jupyter Notebook. A list mapping every figure to the notebook it originated from can be found in table 6.1. The notebooks, as well as the data-subsets used by them are located under `ripeanalysis/jupyter`. Queries, with which we extracted the aforementioned data-subsets can be found under `ripeanalysis/database/SQLite/Queries`.

Jupyter notebook	Figure
Datacenter Lantency comparison.ipynb	4.3 4.4
Geographical Analysis.ipynb	4.1 4.2 4.5 4.6 4.7
IXP peering.ipynb	4.15 4.16 4.17 4.18
Path analysis.ipynb	4.12 4.13 4.14
PSA Study.ipynb	4.8 4.9
TCP vs ICMP.ipynb	4.10a 4.10b 4.11

Table 6.1.: Map of Jupyter notebooks to figures

Package	Version	Package	Version	Package	Version
arrow	0.15.8	attrs	19.3.0	backcall	0.2.0
basemap	1.2.0	bleach	3.1.5	brotlipy	0.7.0
certifi	2020.6.20	cffi	1.14.0	cfu	1.5.0
chardet	3.0.4	click	7.1.2	click-plugins	1.1.1
cligj	0.7.0	cmake	3.18.0	coverage	5.3
cryptography	2.9.2	cycler	0.10.0	decorator	4.4.2
defusedxml	0.6.0	descartes	1.1.0	entrypoints	0.3
Fiona	1.8.17	future	0.18.2	geographiclib	1.50
geopandas	0.8.1	geopy	2.0.0	idna	2.10
importlib-metadata	1.7.0	iniconfig	1.0.1	ipykernel	5.3.3
ipython	7.16.1	ipython-genutils	0.2.0	ipywidgets	7.5.1
jedi	0.17.1	Jinja2	2.11.2	jinja2-time	0.2.0
joblib	0.17.0	jsonschema	3.2.0	jupyter	1.0.0
jupyter-client	6.1.6	jupyter-console	6.1.0	jupyter-contrib-core	0.3.3
jupyter-core	4.6.3	jupyter-nbextensions-configurator	0.4.1	kiwisolver	1.2.0
make	0.1.6.post1	mapclassify	2.3.0	MarkupSafe	1.1.1
matplotlib	3.3.1	mistune	0.8.4	mkl-fft	1.2.0
mkl-random	1.1.1	mkl-service	2.3.0	munch	2.5.0
munge	1.0.0	nbconvert	5.6.1	nbformat	5.0.7
networkit	7.0	networkx	2.5	ninja	1.10.0.post1
notebook	6.0.3	numpy	1.19.1	olefile	0.46
packaging	20.4	pandas	1.1.3	pandocfilters	1.4.2
parso	0.7.0	peeringdb	1.0.0	pexpect	4.8.0
pickleshare	0.7.5	Pillow	7.2.0	pip	20.1.1
pluggy	0.13.1	pprintpp	0.4.0	prometheus-client	0.8.0
prompt-toolkit	3.0.5	ptyprocess	0.6.0	py	1.9.0
pyasn	1.6.0b1	pycountry	20.7.3	pycountry-convert	0.7.2
pycparser	2.20	Pygments	2.6.1	pyOpenSSL	19.1.0
pyparsing	2.4.7	pyproj	2.6.1.post1	pyrsistent	0.16.0
pyshp	2.1.2	PySocks	1.7.1	pytest	6.1.0
pytest-cov	2.10.1	pytest-mock	3.3.1	python-dateutil	2.8.1
pytz	2020.1	PyYAML	5.3.1	pyzmq	19.0.1
qtconsole	4.7.5	QtPy	1.9.0	repoze.lru	0.7
requests	2.24.0	ripe.atlas.cousteau	1.4.2	scikit-learn	0.23.2
scipy	1.5.2	seaborn	0.11.0	Send2Trash	1.5.0
setuptools	49.2.0.post20200714	Shapely	1.7.1	sip	4.19.13
six	1.15.0	socketIO-client	0.7.2	terminado	0.8.3
testpath	0.4.4	threadpoolctl	2.1.0	toml	0.10.1
toolz	0.10.0	tornado	6.0.4	tqdm	4.49.0
traitlets	4.3.3	twentyc.rpc	0.4.0	urllib3	1.25.10
wcwidth	0.2.5	webencodings	0.5.1	websocket-client	0.57.0
wheel	0.34.2	widgetsnbextension	3.5.1	zipp	3.1.0

Table 6.2.: Used Python packages and version

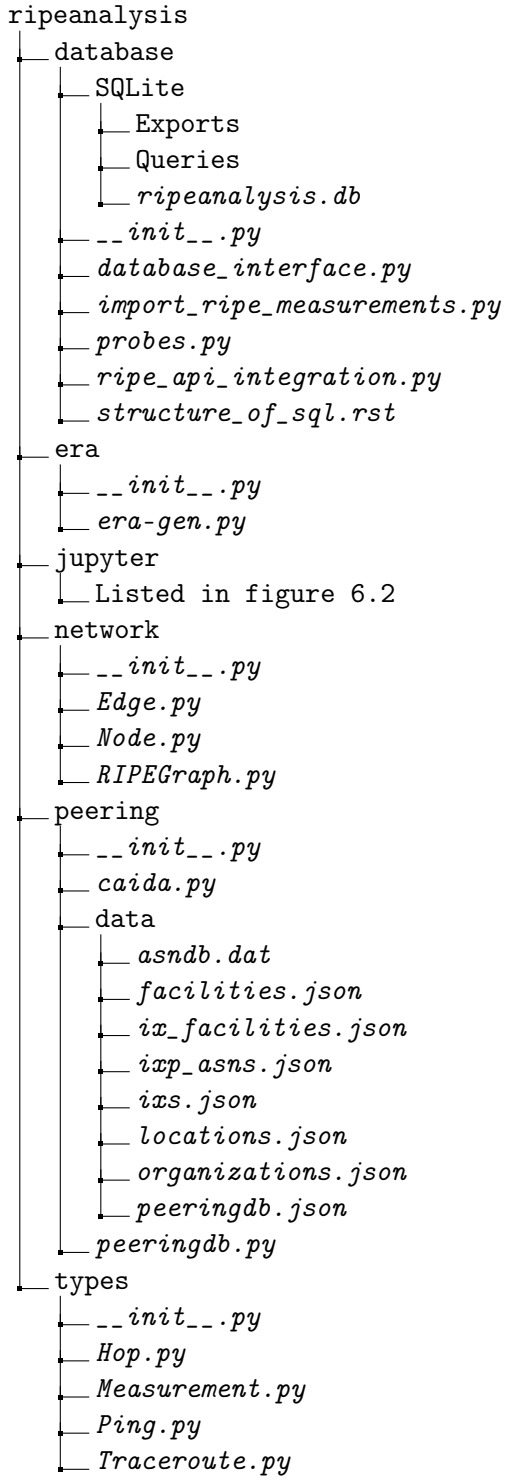


Figure 6.1.: Folder structure of `ripeanalysis` package

```
jupyter
├── all_pings.csv
├── ases
├── continent
├── Datacenter Lantency comparison.ipynb
├── Geographical Analysis.ipynb
├── h_&_a_per_prov.csv
├── hops
├── IXP peering.ipynb
├── ixp_cases
├── ixp_peering_rev.csv
├── lat_world_tcp
├── latency
├── latency_world
├── lowest
├── lowest_icmp_ping_per_country_and_provider.csv
├── lowest_tcp_prov_coun.csv
├── Path analysis.ipynb
├── ping_comp_troute.csv
├── ping_to_closest_dc_intercontinental.csv
├── ping_to_closest_dc_min_only.csv
├── ping_to_closest_dc.csv
├── PSA Study.ipynb
├── psa.csv
├── RIPE Analysis 2.0.ipynb
├── TCP vs ICMP.ipynb
└── us_probes.csv
```

Figure 6.2.: Folder structure of `jupyter` directory

A. Appendix

A.1. RIPE Atlas traceroute JSON structure

A measurement result is a JSON object with the following fields:

- **"af"** – address family, 4 or 6 (integer)
- **"bundle"** – [optional] instance ID for a collection of related measurement results (int)
- **"dst_addr"** – IP address of the destination (string)
- **"dst_name"** – name of the destination (string)
- **"endtime"** – Unix timestamp for end of measurement (int)
- **"from"** – IP address of the probe as know by controller (string)
- **"group_id"** – [optional] If the measurement belongs to a group of measurements, the identifier of the group (int)
- **"lts"** – last time synchronised. How long ago (in seconds) the clock of the probe was found to be in sync with that of a controller. The value -1 is used to indicate that the probe does not know whether it is in sync (int)
- **"msm_id"** – measurement identifier (int)
- **"msm_name"** – measurement type "Traceroute" (string)
- **"paris_id"** – variation for the Paris mode of traceroute (int)
- **"prb_id"** – source probe ID (int)
- **"proto"** – "UDP", "ICMP", or "TCP" (string)
- **"result"** – list of hop elements (array of objects)
  objects have the following fields:
  - **"hop"** – hop number (int)
  - **"error"** – [optional] when an error occurs trying to send a packet. In that case there will not be a result structure. (string)
  - **"result"** – variable content, depending on type of response (array of objects) objects have the following fields:
    Case: Timeout

- **"x"** – "*" (string)

Case: Reply

- **"err"** – (optional) error ICMP: "N" (network unreachable,), "H" (destination unreachable), "A" (administratively prohibited), "P" (protocol unreachable), "p" (port unreachable) "h" (string) Unrecognized error codes are represented as integers
- **"from"** – IPv4 or IPv6 source address in reply (string)
- **"itos"** – (optional) type-of-service (IPv6 traffic class) in the packet that triggered the error ICMP. Omitted if zero and the TOS/Traffic Class field is not set in outgoing packets (int)
- **"ittl"** – (optional) time-to-live in the packet that triggered the error ICMP. Omitted if equal to 1 (int)
- **"edst"** – (optional) destination address in the packet that triggered the error ICMP if different from the target of the measurement (string)
- **"late"** – (optional) number of packets a reply is late, in this case rtt is not present (int)
- **"mtu"** – (optional) path MTU from a packet too big ICMP (int)
- **"rtt"** – round-trip-time of reply, not present when the response is late (float)
- **"size"** – size of reply (int)
- **"ttl"** – time-to-live in reply (int)
- **"flags"** – (optional) TCP flags in the reply packet, for TCP traceroute, concatenated, in the order 'F' (FIN), 'S' (SYN), 'R' (RST), 'P' (PSH), 'A' (ACK), 'U' (URG) (string)
- **"dstoptsize"** – (optional) size of destination options header (IPv6) (int)
- **"hbhoptsize"** – (optional) size of hop-by-hop options header (IPv6) (int)
- **"icmpext"** – [optional] information when icmp header is found in reply (object with the following fields:)
- **"version"** – RFC4884 version (int)
- **"rfc4884"** – "1" if length indication is present, "0" otherwise (int)
- **"obj"** – elements of the object (array of objects).
  objects have the following fields:
  - **"class"** – RFC4884 class (int)
  - **"type"** – RFC4884 type (int)
  - **"mpls"** – [optional] MPLS data, RFC4950, shown when class is "1" and type is "1" (array of objects)
    objects have the following fields:

    - **"exp"** – for experimental use (int)
    - **"label"** – mpls label (int)
    - **"s"** – bottom of stack (int)
    - **"ttl"** – time to live value (int)
    - **"size"** – packet size (int)
- **"src_addr"** – source address used by probe (string)
- **"timestamp"** – Unix timestamp for start of measurement (int)
- **"tos"** – (optional) type-of-service (IPv6 traffic class) in the reply packet. Omitted if zero and the TOS/Traffic Class field is not set in outgoing packets (int)
- **"ttr"** – time to resolve dst_name in milliseconds (float)
- **"type"** – "traceroute" (string)

(Source: `https://atlas.ripe.net/docs/data_struct/`**)**

A.2. RIPE Atlas ping JSON structure

A measurement result is a JSON object with the following fields:

- **"af"** – address family, 4 or 6 (integer)
- **"avg"** – average round-trip time (float)
- **"bundle"** – [optional] instance ID for a collection of related measurement results (int)
- **"dst_addr"** – IP address of the destination (string)
- **"dst_name"** – name of the destination (string)
- **"dup"** – number of duplicate packets (int)
- **"from"** – IP address of the probe as known by the controller (string)
- **"group_id"** – [optional] If the measurement belongs to a group of measurements, the identifier of the group (int)
- **"lts"** – last time synchronised. How long ago (in seconds) the clock of the probe was found to be in sync with that of a controller. The value -1 is used to indicate that the probe does not know whether it is in sync (int)
- **"max"** – maximum round-trip time (float)
- **"min"** – minimum round-trip time (float)

- **"msm_id"** – measurement identifier (int)
- **"msm_name"** – measurement type "Ping" (string)
- **"prb_id"** – source probe ID (int)
- **"proto"** – "ICMP" (string)
- **"rcvd"** – number of packets received (int)
- **"result"** – variable content, depending on type of response (array of objects)
  objects have the following fields:
  Case: Timeout
  - **"x"** – "*" (string)

  Case: Error
  - **"error"** – description of error (string)

  Case: Reply
  - **"rtt"** – round-trip-time in milliseconds (float)
  - **"src_Addr"** – [optional] source address if different from the source address in first reply (string)
  - **"ttl"** – [optional] time-to-live reply if different from ttl in first reply (int)
  - **"dup"** – [optional] signals that the reply is a duplicate (int)
  - **"sent"** – number of packets sent (int)
  - **"size"** – packet size (data part, not including IP and ICMP header) (int)
  - **"src_addr"** – source address used by probe (string)
  - **"timestamp"** – Unix timestamp (int)
  - **"ttl"** – time-to-live field in the first reply (missing due to a bug)(int)
  - **"ttr"** – time to resolve dst_name in milliseconds (float)
  - **"type"** – "ping" (string)

(Source: `https://atlas.ripe.net/docs/data_struct/`**)**

A.3. RIPE Atlas user-tags identified with probe categories

Category	RIPE Atlas User-Tags
home	'freifunk', 'freifunk-haloch', 'freifunk-karlsruhe', 'freifunk-rheinland', 'freifunk-sudpfalz', 'fritzbox', 'guest-lan', 'home', 'home-co-ltd', 'home-lab', 'home-office', 'homelab', 'homeoffice', 'macbook', 'magenta', 'magenta-zuhause', 'pi-hole', 'pihole', 'pihole-2', 'play', 'raspberry-pi-1-mod-b', 'raspberry-pi-1-mod-b-2', 'raspberrypi', 'residential'
organization	'comcast', 'comcast-50x10', 'comcast-business-services', 'comcast-xfinity200mbps10mbps', 'research', 'san-company', 'sandisk', 'santa-fe', 'satellite', 'speedbone', 'speedify', 'student', 'student-dormitories-in-stuttgartgermany', 'student-dormitory', 'student-network', 'studentenwohnheim', 't-mobile', 't-mobile-thuis', 't-online-telekom', 'technicolor', 'tele2', 'telecentro', 'telefonica', 'telekom', 'telenet', 'telenor', 'telfort', 'telia', 'telia-2', 'teliasonera', 'telstra', 'testing', 'testing-tagging-3', 'testlab', 'tiscali', 'tlc', 'tmobile', 'undergraduate', 'unitelnet', 'unitymedia', 'university', 'vodafone', 'vodafone-2', 'vodafone-ita', 'vodafonegigabit'
datacenter	'data-centre', 'datacenter', 'datacentre', 'mobile-datacentre', 'us-central1-a', 'us-east1-b', 'us-east4-c'
ixp	'ix','ixp'

Table A.1.: RIPE Atlas user-tags identified to belong to a certain category of probes

A.4. PeeringDB organization names of cloud providers

PeeringDB organization names
Oracle Cloud Infrastructure
Amazon.com
SoftLayer Technologies Inc. (an IBM Company)
Google LLC
Linode AS63949
DigitalOcean
Microsoft
Alibaba
Choopa, LLC
Alibaba (China)

Table A.2.: PeeringDB organization names of cloud providers

B. Database Excerpt

	ID	name	url	country	continent
1	1	Amazon EC2	ap-southeast-1.ec2.cloudharmony.net	JAP	AS
2	2	Amazon EC2	ap-southeast-2.ec2.cloudharmony.net	AUS	OC
3	3	Amazon EC2	ap-northeast-1.ec2.cloudharmony.net	SGP	AS
4	4	Amazon EC2	ap-northeast-2.ec2.cloudharmony.net	KOR	AS
5	5	Amazon EC2	us-east-1.ec2.cloudharmony.net	USA	NA
6	6	Amazon EC2	us-east-2a.ec2.cloudharmony.net	USA	NA
7	7	Amazon EC2	us-east-2b.ec2.cloudharmony.net	USA	NA
8	8	Amazon EC2	us-west-1.ec2.cloudharmony.net	USA	NA
9	9	Amazon EC2	us-west-2.ec2.cloudharmony.net	USA	NA
10	10	Amazon EC2	af-south-1.ec2.cloudharmony.net	ZAF	AF

Figure B.1.: Datacenter table

	hop_id	src_ip	dst_ip	rtt_before	rtt_after	ttl	hop_num	attempt	Traceroute_ID
1	1	156.38.0.135	156.38.0.140	0	0.386	64	1	0	266334070000
2	2	156.38.0.135	156.38.0.140	0	0.261	64	1	1	266334070000
3	3	156.38.0.135	156.38.0.140	0	0.284	64	1	2	266334070000
4	4	156.38.0.140	196.60.9.105	0.386	0.627	254	2	0	266334070000
5	5	156.38.0.140	196.60.9.105	0.261	1.004	254	2	1	266334070000
6	6	156.38.0.140	196.60.9.105	0.284	0.428	254	2	2	266334070000
7	7	196.60.9.105	52.93.56.40	0.627	1.561	250	3	0	266334070000
8	8	196.60.9.105	52.93.56.40	1.004	1.248	250	3	1	266334070000
9	9	196.60.9.105	52.93.56.40	0.428	1.647	250	3	2	266334070000
10	10	52.93.56.40	52.93.56.53	1.561	0.516	252	4	0	266334070000

Figure B.2.: Hops table

	ip	asn	org_name_pdb	org_type_pdb	is_ixp	ix_id	org_name_caida	region	country	city	latitude	longitude	facility_name
1	168.167.253.33	14988	Botswana Telecommunications Corporation	Not Disclosed	0	533	NAPAfrica IX Johannesburg	Africa	ZA	unknown	200	200	Teraco Johannesburg Campus, South Africa
2	41.212.1.65	15399	Wananchi Group	Cable/DSL/ISP	0	437	London Internet Exchange	Europe	GB	London	51.48	-0.45	no facility
3	150.222.29.157	NULL	Unknown ASN	Unknown	0								
4	168.209.86.217	3741	Internet Solutions, South Africa	NSP	0	533	NAPAfrica IX Johannesburg	Africa	ZA	unknown	200	200	Teraco Johannesburg Campus, South Africa
5	150.222.94.140	NULL	Unknown ASN	Unknown	0								
6	150.222.29.120	NULL	Unknown ASN	Unknown	0								
7	102.132.135.239	37680	Cool Ideas Service Provider Pty Ltd	Cable/DSL/ISP	0	581	New York International Internet Exchange	North America	US	New York	40.75	-74	Digital Realty NYC (60 Hudson)
8	196.41.97.21	36874	Cybersmart Ltd	Cable/DSL/ISP	0	533	NAPAfrica IX Johannesburg	Africa	ZA	unknown	200	200	Teraco Johannesburg Campus, South Africa
9	41.222.1.49	36997	Infocom Ltd	Cable/DSL/ISP	0	808	Uganda Internet Exchange	Africa	UG	Kampala	0.32	32.62	no facility
10	41.203.255.222	36902	Intelvision	Cable/DSL/ISP	0	437	London Internet Exchange	Europe	GB	London	51.48	-0.45	no facility

Figure B.3.: NodeInfo table

	ID	MSMID_Pin	platform_	protocol	timestamp	url	src	dst	probe_id	ttl	ping1	ping2	ping3	ping4	ping5
1	1	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	156.38.0.135	13.244.117.236	1000070	242	21.208011	21.208991	21.156998	0	0
2	2	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	196.61.64.64	13.244.117.236	1000237	239	22.065525	21.92665	21.958963	0	0
3	3	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	160.242.14.157	13.244.117.236	1000327	238	70.985074	64.278445	74.477537	0	0
4	4	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	41.79.219.203	13.244.117.236	1000468	227	337.128921	363.951329	387.426134	0	0
5	5	26801536	ripe	ICMP	1598287523	af-south-1.ec2.cloudharmony.net	137.255.6.37	13.244.117.236	1000469	234	266.830504	267.024035	266.906299	0	0
6	6	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	197.234.221.42	13.244.117.236	1000470	227	276.128845	281.757268	279.259888	0	0
7	7	26801536	ripe	ICMP	1598287523	af-south-1.ec2.cloudharmony.net	160.119.236.94	13.244.117.236	1000484	237	2.640513	2.294038	2.236063	0	0
8	8	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	169.255.0.135	13.244.117.236	1000492	241	17.951581	17.797435	17.805451	0	0
9	9	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	13.244.74.202	13.244.117.236	1000707	254	0.127969	0.140455	0.132001	0	0
10	10	26801536	ripe	ICMP	1598287524	af-south-1.ec2.cloudharmony.net	160.119.228.175	13.244.117.236	1000825	237	27.7295	26.5719	25.9267	0	0

Figure B.4.: Ping table

	ID	country	continent	home	organizat	datacente	ixp	longitude	latitude
1	1	NLD	EU	1	0	0	0	4.9275	52.3475
2	2	NLD	EU	1	0	0	0	4.9575	52.3085
3	3	NLD	EU	1	0	0	0	4.9375	52.3685
4	4	NLD	EU	1	0	0	0	4.6375	52.3895
5	5	NLD	EU	1	0	0	0	4.9175	52.0595
6	6	NLD	EU	1	0	0	0	4.9175	52.3505
7	7	NLD	EU	0	0	0	0	6.0075	51.2005
8	8	NLD	EU	1	0	0	0	6.0375	51.2315
9	9	NLD	EU	0	0	0	0	4.8975	52.3815
10	10	NLD	EU	1	0	0	0	4.9175	52.3475

Figure B.5.: Probes table

	ID	MSMID_Tra	platform_	protocol	timestamp	src	dst	paris_id
1	266334070000	26633407	ripe	ICMP	1596916637	156.38.0.135	13.244.117.236	0
2	266334070001	26633407	ripe	ICMP	1596916637	196.61.64.64	13.244.117.236	0
3	266334070002	26633407	ripe	ICMP	1596916637	160.242.14.157	13.244.117.236	0
4	266334070003	26633407	ripe	ICMP	1596916637	197.155.17.215	13.244.117.236	0
5	266334070004	26633407	ripe	ICMP	1596916637	169.255.0.135	13.244.117.236	0
6	266334070005	26633407	ripe	ICMP	1596916636	13.244.74.202	13.244.117.236	0
7	266334070006	26633407	ripe	ICMP	1596916637	196.170.57.252	13.244.117.236	0
8	266334070007	26633407	ripe	ICMP	1596916636	41.221.32.130	13.244.117.236	0
9	266334070008	26633407	ripe	ICMP	1596916638	41.220.0.39	13.244.117.236	0
10	266334070009	26633407	ripe	ICMP	1596916636	196.1.95.103	13.244.117.236	0

Figure B.6.: Traceroute table

	ID	probe_id	cloud_provider	datacenter	url
1	266334070000	1000070	Amazon.com	10	af-south-1.ec2.cloudharmony.net
2	266334070001	1000237	Amazon.com	10	af-south-1.ec2.cloudharmony.net
3	266334070002	1000327	Amazon.com	10	af-south-1.ec2.cloudharmony.net
4	266334070003	1000484	Amazon.com	10	af-south-1.ec2.cloudharmony.net
5	266334070004	1000492	Amazon.com	10	af-south-1.ec2.cloudharmony.net
6	266334070005	1000707	Amazon.com	10	af-south-1.ec2.cloudharmony.net
7	266334070006	10203	Amazon.com	10	af-south-1.ec2.cloudharmony.net
8	266334070007	10243	Amazon.com	10	af-south-1.ec2.cloudharmony.net
9	266334070008	10843	Amazon.com	10	af-south-1.ec2.cloudharmony.net
10	266334070009	11383	Amazon.com	10	af-south-1.ec2.cloudharmony.net

Figure B.7.: TracerouteInfo table

List of Figures

List of Tables

Bibliography

[1] A. Li, X. Yang, S. Kandula, and M. Zhang. "CloudCmp: comparing public cloud providers". In: *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 2010, pp. 1–14.

[2] Amazon. *Web Archive: Amazon Elastic Compute Cloud*. `https://web.archive.org/web/20110102062417mp_/http://aws.amazon.com/ec2`. Accessed on 2020-11-09. 2010.

[3] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill. "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity". In: *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*. 2004, pp. 39–47.

[4] N. Mohan, L. Corneo, A. Zavodovski, S. Bayhan, W. Wong, and J. Kangasharju. "Pruning Edge Research with Latency Shears". In: ().

[5] T. Arnold, J. He, W. Jiang, M. Calder, I. Cunha, V. Giotsas, and E. Katz-Bassett. "Cloud Provider Connectivity in the Flat Internet". In: *ACM Internet Measurement Conference (IMC '20)*. 2020, p. 17.

[6] T. Inc. *Cloud Performance Benchmark 2019–2020 Edition*. Tech. rep. ThousandEyes, 2020.

[7] B. T. W. Group et al. "Interconnection and Traffic Exchange on the Internet". In: *Interconnection and Traffic Exchange on the Internet (November 1, 2014)* (2014).

[8] P. Faratin, D. D. Clark, S. Bauer, W. Lehr, P. W. Gilmore, and A. Berger. "The growing complexity of Internet interconnection". In: *Communications & strategies* 72 (2008), p. 51.

[9] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. "There is more to IXPs than meets the eye". In: *ACM SIGCOMM Computer Communication Review* 43.5 (2013), pp. 19–28.

[10] V. Giotsas, G. Smaragdakis, B. Huffaker, M. Luckie, and K. Claffy. "Mapping peering interconnections to a facility". In: *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. 2015, pp. 1–13.

[11] T. Arnold, E. Gürmeriçliler, G. Essig, A. Gupta, M. Calder, V. Giotsas, and E. Katz-Bassett. "(How Much) Does a Private WAN Improve Cloud Performance?" In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 2020, pp. 79–88.

[12] R. N. Staff. "Ripe atlas: A global internet measurement network". In: *Internet Protocol Journal* 18.3 (2015).

[13] V. Bajpai, S. J. Eravuchira, and J. Schönwälder. "Lessons learned from using the ripe atlas platform for measurement research". In: *ACM SIGCOMM Computer Communication Review* 45.3 (2015), pp. 35–42.

[14] P. Gigis, V. Kotronis, E. Aben, S. D. Strowes, and X. Dimitropoulos. "Characterizing user-to-user connectivity with ripe atlas". In: *Proceedings of the Applied Networking Research Workshop*. 2017, pp. 4–6.

[15] J. Gedeon, S. Zengerle, S. Alles, F. Brandherm, and M. Mühlhäuser. "Sunstone: Navigating the Way Through the Fog". In: *2020 IEEE 4th International Conference on Fog and Edge Computing (ICFEC)*. 2020, pp. 49–58.

[16] R. Rajabiun and F. McKelvey. "Complementary realities: Public domain Internet measurements in the development of Canada's universal access policies". In: *The Information Society* 35.2 (2019), pp. 81–94.

[17] X. Deng, Y. Feng, H. H. Gharakheili, and V. Sivaraman. "Estimating Residential Broadband Capacity using Big Data from M-Lab". In: *arXiv preprint arXiv:1901.07059* (2019).

[18] M-Lab. *Internet Measurement Tests*. `https://www.measurementlab.net/tests/`. Accessed on 2020-10-13.

[19] M-Lab. *M-LAb Network Diagnostic Tool*. `https://www.measurementlab.net/tests/ndt/`. Accessed on 2020-10-13.

[20] PlanetLab. *PlanetLab History*. `https://www.planet-lab.org/?q=history`. Accessed on 2020-11-02.

[21] C. Dovrolis, K. Gummadi, A. Kuzmanovic, and S. D. Meinrath. *Measurement lab: Overview and an invitation to the research community*. 2010.

[22] Speedchecker. *ProbeAPI Speedchecker*. `https://probeapi.speedchecker.com`. Accessed on 2020-10-15.

[23] J. Chavula, A. Phokeer, A. Formoso, and N. Feamster. "Insight into Africa's country-level latencies". In: *2017 IEEE AFRICON*. IEEE. 2017, pp. 938–944.

[24] A. Formoso, J. Chavula, A. Phokeer, A. Sathiaseelan, and G. Tyson. "Deep diving into africa's inter-country latencies". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE. 2018, pp. 2231–2239.

[25] RIPE NCC. *Ripe Atlas*. `https://atlas.ripe.net`. Accessed on 2020-10-10.

[26] CloudHarmony. *CloudHarmony Network Test*. `https://cloudharmony.com/speedtest`. Accessed on 2020-10-10.

[27] L. C. et al. "ERA: Enhanced RIPE Atlas Measurement Tool". In: *Submitted for review to ARNW 2020*. 2020.

[28] CAIDA. *CAIDA dataset (202007)*. `http://data.caida.org/datasets/ixps`. Accessed on 2020-11-07.

[29] J. P. van Best. *Unraveling Internet Infrastructure*. Eburon Publishers, 2005.

[30] Wikipedia. *Statistical area (United States)*. `https://en.wikipedia.org/wiki/Statistical_area_United_States`. 2019.

[31] Executive Office of the President, Washington, D.C. 20503. *OMB BULLETIN NO. 13-01*. `https://obamawhitehouse.archives.gov/sites/default/files/omb/bulletins/2013/b13-01.pdf`. Feb. 2013.