Master's Thesis

Anycast-based DNS in Mobile Networks

Graduation Studies conducted for obtaining the Master's degree in Computer science by Sarah Anne Wassermann

NU advisor: Prof. Fabián E. Bustamante
ULg advisor: Prof. Benoit Donnet

Academic year 2016 – 2017

University of Liège – Faculty of Applied Sciences

Contents

List of Figures

List of Tables

Abstract

Anycast offers a method for making a service IP address available to a routing system from several locations at once. It is used today to provide important services, such as naming and content delivery, in an economic, scalable, and simple to operate manner. The appeal and clear benefits of anycast to service providers have motivated a number of recent experimental studies on its potential performance impact. All studies have, to the best of our knowledge, focused on wired networks, despite the growing dominance of mobile as the most common and sometimes only form of Internet access. In this thesis, we present the first study of anycast performance for mobile users. In particular, our evaluation focuses on three distinct anycast services: K- and F-Root, each providing part the DNS root zone, and Google DNS.

Our research revolves around three axes. *First*, we show that mobile clients are frequently routed to sub-optimal replicas in terms of latency and that this issue is not limited to specific regions or ASes of the world. *Second*, we find that clients are often redirected to a DNS server hosted very far away from her. This happens more frequently while on a cellular connection than on WiFi, with a significant impact on performance. Our study reveals that this is not simply an issue of not having better alternatives, and that the problem is not localised to particular geographic areas or particular ASes. We investigate root causes of this phenomenon and describe three of the major detected classes of anycast anomalies. *Third* and finally, we explore IP assignment dynamics of mobile clients and find that recurrent IP changes on the client side lead to significant perceived variations of anycast latency.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my MSc. research advisor, Prof. Fabián Bustamante, for having given me the opportunity to work on this exciting topic and having hosted me in the Aqualab group during summer 2016. Throughout our collaboration, you have motivated me to dive deeper and deeper into our research questions, and your guidance is an invaluable help. I will never forget this enriching experience.

I also would like to thank Dr. John Rula, who was not only a nice lab mate, but also a great mentor during my research stay at Northwestern University. Thanks John for all your work on our measurement campaign and for our discussions about research and the life of a PhD student.

I had the chance to have Prof. Benoit Donnet as a local MSc. advisor. I am not only grateful to him for his guidance in the context of this Master's thesis, but also because he has been supporting me since my first year at the University of Liège. Most importantly, he helped me organise my first two internships. My research journey would never have been the same without you, thank you.

Even though he was not directly involved in this thesis, I cannot write acknowledgements without mentioning Dr. Pedro Casas. Since summer 2015, I have been working with Pedro on multiple research projects and I performed two research internships under his supervision. He is also the one who introduced me to Fabián. However, Pedro is not only an exceptional researcher and supervisor, but also a fantastic human being. Thanks to you, I discovered the beauty of research in the field of Internet measurements and got addicted to it. Our collaboration and insightful discussions helped me grow as a human, and overcome some very hard times. Pedro, I am forever grateful for everything you have done for me.

Next, I want to express a special thank you to my mother. Throughout my whole life, you supported me when I was about to give up. During the last five years, you suffered with me when the time for exams had come, and tried to help me whenever you could to make my student life more enjoyable.

Last but not least, I want to thank Thibaut Cuvelier for his unconditional love. Thank you for always believing in me and helping me whenever possible. My conference trips and my last year here in Liège are unforgettable thanks to everything we have experienced together. Finally, this thesis would not be of the same quality without your countless revisions.

Chapter 1

Introduction

Service providers rely on replication to improve service performance and reliability, placing server instances in multiple locations and redirecting clients to nearby copies of the requested data. IP anycast is a common mechanism used for redirecting clients in a variety of domains: from naming (e.g., root servers, top level domains, and many public resolvers) to CDNs and video streaming.

With IP anycast, services advertise a single IP address from many physical locations and clients' requests are directed, based on BGP routing policies, to a "nearby" replica [1].

From an operator's perspective, IP anycast offers an economic, scalable, and simple approach to replicated services, as BGP routing provides considerable robustness, adapting to changes in service and network availability. From a client's perspective, however, the mapping can be suboptimal [2], unstable [3, 4], and seemingly chaotic [5, 6], as routing policies have not only technical motivations, but could also be dictated by political or commercial reasons, and routing changes can silently shift traffic from one site to another with a consequent loss of shared state and potential performance impact [7].

Given its wide deployment, criticality, and interesting trade-offs, IP anycast has been the focus of much recent measurement work. All prior studies have, however, focused on wired networks, despite the growing dominance of mobile as the most common form of Internet access.

The number of mobile subscriptions has grown rapidly in just a few years, surpassing 7.4 billion in the first quarter of 2016 [8]. Today, users spend most of their time browsing on their mobile phones, more than on any other device. In the United States and the United Kingdom, for instance, smartphone users spend 2-3x more hours (87 hours in the US and 66 hours in the UK) on their mobile device than on desktop machines (34 hours and 29 hours, respectively) [9]. The impressive growth in mobile devices and usage has led to unprecedented growth in cellular traffic.

In this Master's thesis, we present the first study of anycast performance for mobile users. In particular, our evaluation focuses on three distinct anycast services: K-Root, F-Root, and Google DNS. F- and K-Root provide part the DNS root zone, unlike Google DNS. We show that mobile clients are routed to suboptimal replicas in terms of latency and geographic distance. Our study reveals that this is not simply an issue of not having better alternatives, and that the problem is not localised to particular geographic areas or particular ASes. We begin to explore the potential root causes of the geographical distance phenomenon in cellular networks and reveal three major classes of anycast anomalies. Additionally, we look at the IP assignment dynamics on mobile networks. We find that cellular clients change their IP address much more frequently than WiFi users. This peculiarity is most probably linked to their greater mobility, and has a significant impact on perceived latency variations for the three anycast services.

This thesis is structured as follows. The Chapters 2 to 5 aim at providing the necessary background to fully understand the research detailed in this document. In particular, these chapters introduce the anycast paradigm (Chapter 2), cellular networks (Section 3), the domain name system (DNS) and its use of anycast (Chapter 4), and finally the currently most used Internet diagnostic tools, `ping` and `traceroute` (Chapter 5). Chapters 6 to 9 present our work. Chapter 6 explains our measurement campaign. In Chapter 7, we show that the performance of anycast is often suboptimal in terms of latency (i.e. another replica would provide smaller latency), independently of the considered DNS service. Our findings described in Chapter 8 demonstrate that mobile clients are very often routed to a suboptimal server in terms of geographic distance and we analyse potential root causes. In Chapter 9, we take a look at IP assignment dynamics, i.e. how many different IPs a client gets assigned over time to and how frequently she changes from one IP to another. Chapter 10 concludes this work.

Chapter 2

Anycast

Summary. This thesis focuses on anycast in cellular networks. In this chapter, we first explain in Section 2.1 the unicast scheme, the currently most used one in the Internet. In Section 2.2, we present the anycast paradigm and why it is so attractive for providers. In Section 2.3, we discuss related work.

2.1 Unicast Paradigm

Unicast is the most widely deployed communication scheme. In this context, a transmission only involves one specific receiver and one sender. Each IP is assigned to one machine at a time. This implies that every packet with the same receiver IP is routed to the same machine, independently of the location of the sender. The unicast routing scheme is illustrated in Figure 2.1.

This paradigm might cause performance troubles, as the client may have to contact a server very far from her.

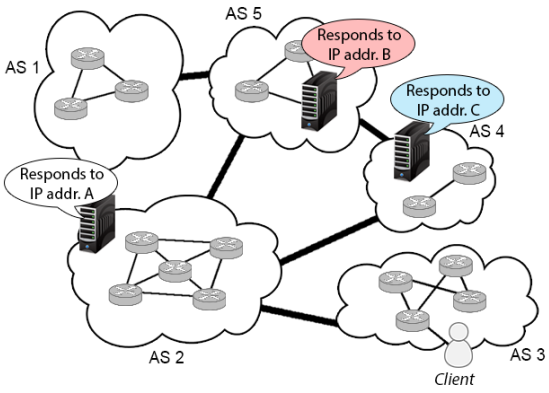


Figure 2.1: Unicast routing scheme.

2.2 Anycast Paradigm

Anycast offers a method for making a service IP address available to a routing system from several locations at once. This means that, contrary to the unicast scheme, several (mostly geographically distributed, but not necessarily) machines, so-called *replicas*, respond to the same IP address [10]. More precisely, an anycast service uses several *sites*, i.e. several locations hosting replicas, to provision its clients. We define as *catchment* of a site the users served by this site [2].

The anycast scheme is popular among multiple services. For instance, LinkedIn, root DNS services, or Content Delivery Networks (CDNs) such as CloudFlare heavily rely on this methodology. The way a user is assigned to a server differs between the different variations of the anycast paradigm, which are explained in Sections 2.2.1 and 2.2.2.

2.2.1 IP-layer anycast

In the context of IP-layer anycast, also called *L4 anycast*, the server selection depends on BGP. In a nutshell, the same IP address (i.e. the anycast IP address) is advertised across many locations thanks to BGP announcements. The server the client is assigned to is selected based on the best-path notions of BGP, such as shortest AS-path length, Hot Potato criterion, etc. [1]. Figure 2.2 illustrates an example of an IP-layer anycast application.

However, as pointed out in [11], IP-layer anycast faces some challenges. One major issue is that L4 anycast (as well as BGP) does not take into account the dynamics of the network, such as quality changes along the different paths and server load. Moreover, L4 anycast can lead to the interruption of ongoing TCP connections, which then need to be restarted. Obviously, this results in a performance penalty for the affected users.

In this thesis, we focus on this anycast paradigm.

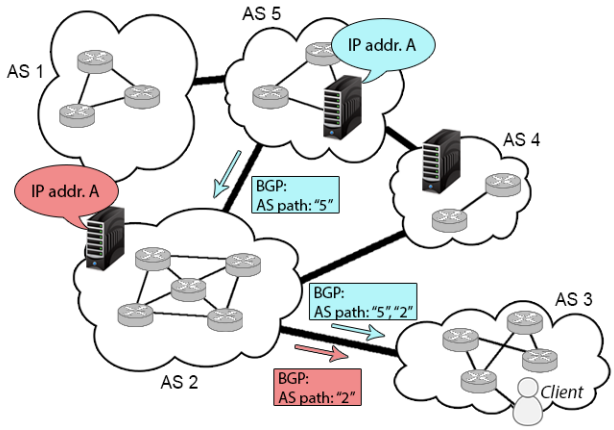


Figure 2.2: L4 anycast. The selection of the server for the client is carried out based on BGP announcements. In this example, AS 3 will choose one of the two anycast servers based on BGP's best-path notions.

Autonomous System (AS)
An Autonomous System regroups multiple routers, which are thus often under the same administrative control. Each AS is identified by a unique *Autonomous System Number* (ASN). Popular ASes are for example Google's AS, AS15169, and the AS managed by Comcast, AS7922.
An *AS path* corresponds to all the ASes a packet traverses to reach its destination.

Border Gateway Protocol (BGP)
BGP [12] is the most used (and most complex) routing protocol in today's Internet. If the maintainers of an AS decide that it should be reachable from the outside, they need to ensure that the other ASes are aware of its existence. This is where BGP comes into play.
In a nutshell, this protocol works as follows: *gateway routers*, i.e. routers that lie at the border of a network, send a message to neighbouring gateway routers containing the prefixes hosted within their network. The gateway router receiving the message distributes this information among the routers in this AS, so that the routers can update their routing tables accordingly. The gateway then appends the prefixes included in its own network to the list of received prefixes and advertises the updated list to its own neighbour networks, and so on.
However, these BGP messages also contain the ASes they have already traversed, i.e. an AS path. Before a gateway router distributes the list of prefixes to the routers, it therefore first checks whether its ASN is in the AS path of the message. If so, it has experienced a loop in the routing system and is simply dropped.

2.2.2 Application-layer anycast

Application-layer anycast, also called *L7 anycast*, heavily relies on DNS redirection and IP unicast. The server-selection procedure is depicted in Figure 2.3. As explained in [11], this scheme works as follows.

First, the user requests a resource, such as a video or a HTML webpage, via its URL. The user agent infers from this URL the hostname, probably belonging to a certain Content Delivery Network. In this case, the

main role is then played by the client's local DNS resolver and the CDN: once the DNS resolver receives the hostname the user wants to contact, it forwards this hostname to the concerned CDN.

Next, the CDN chooses the node that should serve the client. This can be achieved in several ways [13]. For instance, the CDN simply chooses the geographically closest node to the client. Another approach is to select the closest CDN node in terms of latency with respect to the client, or the closest in terms of number of hops. To this end, CDN servers periodically launch `ping` or `traceroute` measurements towards access Internet Service Providers (ISPs), and report the results to the DNS resolvers of the CDN (more details about `ping` and `traceroute` can be found in Chapter 5). The CDN may also return the *same* IP address, independently of the user. In this case, the IP address is an IP-layer-anycast address (see Section 2.2.1), associated to several (or even all) CDN nodes to balance the load among the different servers. The main issue with these methods is that the client is represented by her DNS resolver, which may be relatively far away from the user and thus does not necessarily represent her very well [14]: the server selection might therefore be suboptimal in terms of performance for the user. Alternatively, this selection can be performed on the *client* side: a possible technique is that the manifest file of the resource to fetch contains a series of IP addresses and that the client chooses the optimal one based on latency, i.e. by pinging the servers indicated in the manifest file. This solution is implemented by Netflix [13].

As a final step, the DNS resolver forwards the selected IP address to the client.

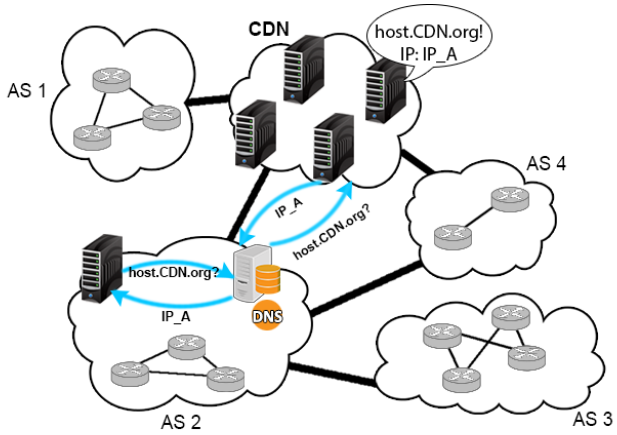


Figure 2.3: L7 anycast. The client wants to know the IP address mapped to the hostname `host.CDN.org`. The DNS resolver forwards the request to the CDN to get the IP address of the "best" server for this client.

2.3 Related Work

Anycast plays a central role in computer networks and is thus subject to many research efforts, dealing with different aspects of the paradigm.

2.3.1 Anycast in Content Delivery Networks

A large part of previous work [15, 16, 17, 18, 19, 14, 20, 11] aims at characterising the server deployment of popular services relying on either L7 or L4 anycast.

[18, 19, 15] focus on the characterisation of YouTube's server infrastructure and the geolocation of these servers [15]. In [18], the authors study the characteristics of the servers seen from both fixed-line and mobile networks. They find that caching significantly enhances the performance in terms of delay and downlink throughput for mobile clients, whereas there is a non-negligible performance variation for fixed-line users. [19] presents the impact of the server-selection strategies employed by YouTube on the Quality of Experience (QoE) of the users.

[16, 20, 17] study the server deployment of Netflix. They find that Netflix heavily relies on Amazon cloud services and on three different CDNs: Akamai, Limelight, and Level-3. Adhikari et al. also reveal that the CDN selection is rather static, i.e. a player sticks to the same CDN, even if another could provide better QoE. [17] additionally sheds light on the location of the different Netflix servers. According to the authors, approxim-

ately 5,000 servers are located in at least 243 different locations across six continents, with a concentration in North America and Europe.

Authors of [14] assess the L7-anycast strategy used in a Akamai's CDN. They state that most of the clients are located near their local DNS resolvers, which implies that L7 anycast performs well in most cases. However, some DNS resolvers serve clients scattered across a larger geographical region, and the authors show that these clients experience performance issues with L7 anycast. They also present a redirection approach based on EDNS-Client-Subnet (ECS) [21] and achieve significant performance improvements.

[11] analyses the performance implications of L4 anycast in a latency-sensitive CDN. More precisely, they study the Microsoft Bing search service. They find that L4 anycast works fairly well, but that 20% of the clients are redirected to a suboptimal server in terms of performance. In this thesis, we are studying a similar problem.

Alzoubi et al. present in [22, 23] a load-aware CDN architecture, which makes use of route control mechanisms to take server and network load into account to address the L4-anycast issues explained in Section 2.2.1. Authors of [24] showcase *FastRoute*, a similar system to [22, 23], but with an emphasis on scale.

2.3.2 Anycast in DNS services

We are not the first ones investigating anycast in DNS. A lot of related work already deals with anycast within DNS services [25, 26, 27, 28, 29, 30, 6, 31]. The main focus of these research efforts are server enumeration [25] and characterisation [25, 26, 27, 28, 29, 6, 31]. When it comes to the analysis of more specific aspects in anycast, we can subdivide prior work in several categories: studies of the proximity between the users and the selected servers [26, 27, 29]; research concerned with user affinity, i.e. how many different anycast servers a client gets directed to over time and how frequently she switches between these servers [27, 28, 29, 31]; work about anycast load distribution[29, 28]; and, lastly, research studying anycast-server availability [29, 31].

In [6], the author concludes that about half of the vantage points (VPs) he uses in the context of his evaluation are served by a suboptimal server of the K-Root DNS infrastructure, both in terms of latency and geographical distance.

While [6, 26] study K-Root, authors of [27, 29, 28, 5] primarily focus on F- and J-Root server characterisation. The authors acknowledge that the use of anycast improves DNS service to clients worldwide. In [5], Bellis et al. describe how they make use of RIPE Atlas to detect and address issues related to F-Root. For instance, they notice that packets issued from RIPE Atlas probes in Europe are redirected to a F-Root instance in Atlanta instead of one located in European countries, which they attribute to BGP misconfiguration. Besides F- and J-Root servers, [29] analyses the behaviour of L4 anycast in the Autonomous System (AS) 112, an AS designed for distributing the load across the Internet for a certain type of queries[1]. It is worth mentioning that the latter paper carries out the study at a larger scale as opposed to other work, involving about 20,000 vantage points.

The research presented in [30] is motivated by the potential Distributed Denial of Service (DDoS) attack against many of the root DNS servers on November 30 and December 1, 2015 (the traffic directed towards a large number of them was multiplied by a factor 100 with respect to normal load). In this paper, besides analysing data collected during these two days, Moura et al. also evaluate L4 anycast under stress with public data.

2.3.3 Anycast for mobile performance improvement

Nowadays, a significant amount of data traffic is generated by mobile devices. Researchers therefore aim at optimising the performance of the mobile networks. One aspect they are looking at is anycast [32, 33, 34].

In [34], Chen et al. argue that anycast service discovery in Mobile Ad hoc Networks (MANETs) is challenging and implies large traffic overhead. They thus propose an anycast scheme which reduces the number of necessary queries.

Similarly, [33] presents MQAR, a Mobility- and Quality of Service-aware Anycast routing scheme in MANETs.

The research in [32] describes an anycast-based server selection model that enables roaming mobile clients to detect and access an anycast server that gives better performance based on the mobile clients' latest network location. Indeed, mobile IPv6 imposes its users to remain connected to the same server, even if they are

[1] `https://www.as112.net/`

roaming into foreign networks, in which there might be servers that would provide better service to the clients [35].

2.3.4 Anycast server enumeration

Several other pieces of previous work focus on developing techniques for anycast-server enumeration [36, 37, 38] and geolocation based on latency measurements [36, 37].

Schmidt et al. analyse how many anycast sites shall be used to ensure that all the clients experience satisfactory latency [2]. The authors conclude that increasing the number of sites does not solve the suboptimal mapping issues. Indeed, the authors observe that having more sites does not help lower latency, which does not seem intuitive.

In [39], authors conduct the first Internet-wide anycast census and show that major players in the Internet ecosystem are relying on anycast, even though only a small part of the IPv4 space has so far been anycasted.

To the best of our knowledge, all studies have so far focused on the performance of anycast in fixed-line networks. In this thesis, we carry out the first analysis of the performance of DNS anycast on mobile – cellular and WiFi – networks. In particular, we mainly focus on the K-Root and F-Root DNS services, but also take a look at Google DNS.

Chapter 3

Cellular Networks

Summary. In this thesis, we explore anycast for DNS mostly in cellular networks. This chapter provides a brief overview of this kind of network. First, we will focus on the architecture of a cellular network by detailing the structure of a 2G network (Section 3.1), before moving on to 3G (Section 3.2) as well as 4G technologies (Section 3.3), both building on the grounds laid by 2G (GSM). Mobile Internet performance is a critical point for smartphone users. Unfortunately, the mobile clients' experience is often rather unsatisfactory. In Section 3.4, we take a brief look at the factors influencing the performance of the mobile Internet.

3.1 2G Cellular Network Architecture

2G networks were exclusively designed for voice traffic, as data traffic did not play an important role when the first GSM networks appeared. Figure 3.1 depicts the architecture of such a network.

First of all, the geographic region to be covered is subdivided into so-called *cells* (hence the name of *cellular network*). Each cell contains a set of mobile clients and a *base transceiver station* (BTS), which is responsible for transmitting and receiving signals to and from the mobile devices located in the given cell. A major challenge in wireless networks is to increase the coverage of a cell. Indeed, the latter is influenced by numerous factors, among which are the transmission power of the BTS, the buildings in this area, etc.

Second, every 2G network includes *base station controllers* (BSC) used by multiple BTSes. Indeed, a BSC is responsible for allocating the appropriate BTS channel resources needed by the different mobile clients to communicate. In order to do so, a BSC performs *paging*, i.e. determines for each client in which cell she is located. Furthermore, a BSC takes care of the handoff of clients, which occurs when they switch to another BTS. The base station controller along with all the base transceiver stations under its control form a *base station system* (BSS).

Finally, a 2G network includes at least one *mobile switching center* (MSC). A typical MSC serves approximately five BSSes; its main responsibility is user authentication. They also handle handoff between several BSCs or MSCs. Specific MSCs, called *gateway MSCs*, connect the GSM clients to the larger public telephone network.

3.2 3G Cellular Network Architecture

With the ever growing need to have access to the Internet on the go, 3G made its appearance. In addition to voice traffic, 3G networks explicitly support data traffic. Figure 3.2 shows the scheme of such a network.

The architecture of a 3G network heavily relies on the 2G network structure: the whole 2G architecture is left untouched; the necessary infrastructure is simply grafted onto the 2G network. More precisely, the 3G part is connected to the initially called BSC, which is referred to as *radio network controller* (RNC) in the 3G world.

The 3G core network connects the mobile subscribers to the public Internet. To do so, it is composed of *serving GPRS support nodes* (SGSN) and *gateway GPRS support nodes* (GGSN). The task of a SGSN is to transmit datagrams to/from mobile clients located in the radio access network the SGSN is assigned to. The GGSN is considered as a gateway and connects multiple SGSNs to the public Internet. As such, the GGSN is the last 3G

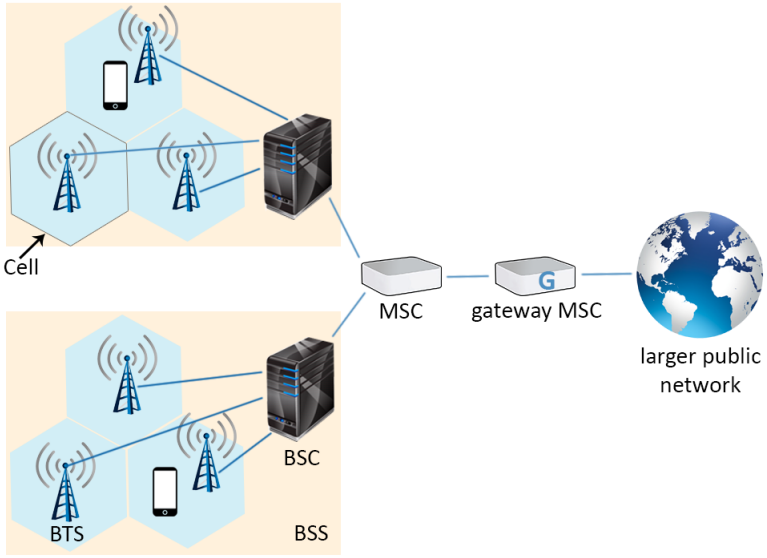


Figure 3.1: Scheme of a 2G network.

architecture component a datagram sent by the mobile user encounters before entering the public Internet. It is also worth mentioning that the GGSN appears to the outside world as a normal gateway router and the mobility dynamics of the 3G protagonists are therefore well hidden.

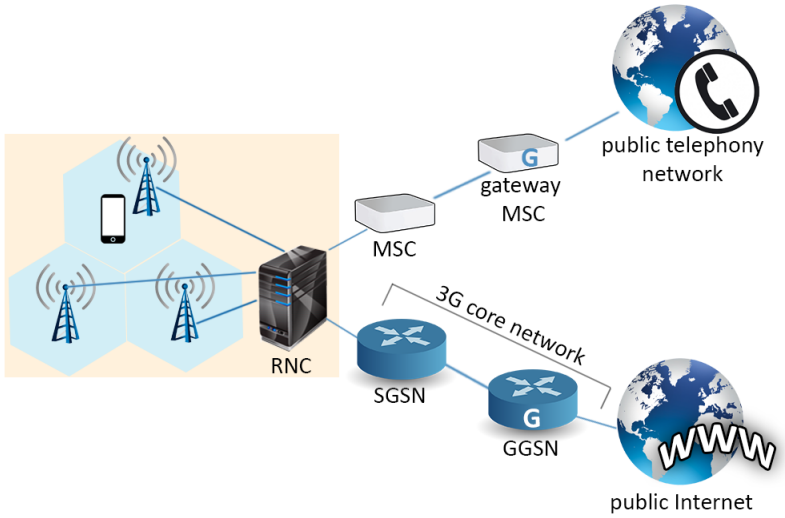


Figure 3.2: Architecture of a 3G network.

3.3 4G Cellular Network Architecture

The aim of the 4G networks was to enhance the performance perceived by the mobile clients (both in terms of latency and throughput).

A 4G network has roughly the same architecture than a 3G one. However, two key points make 4G having a great advantage over 3G networks.

First, 4G does not include a separate circuit-switched network for voice and a packet-switched cellular network for data traffic. Instead, both parts are merged into an IP network, which has as the consequence that both kinds of traffic are transferred using IP datagrams. A very important component of a 4G (LTE) network is

the *Packet Gateway* (PGW), which is the point of contact with the outside world (similar to the GGSN in a 3G architecture). This new kind of network lead to the development of Voice over Long-Term Evolution (VoLTE), a standard to transfer voice packets over 4G networks.

Second, the 4G radio access network allows the usage of multiple input, multiple output (MIMO) antennas, which makes it possible to significantly increase the data rate for mobile clients (up to 100 Mbps for downloads and 50 Mbps for uploads).

3.4 Mobile Internet Performance

Mobile Internet is known to provide very variable experience in terms of performance to its users. Multiple factors influence the performance perceived by a mobile client.

First of all, it is heavily influenced by the signal strength captured by the users' phone, as a longer distance between the phone and its assigned Base Transceiver Station and/or physical objects weaken the signal.

Moreover, the performance depends also on the used network technology: while 4G offers a bandwidth of maximum 100 Mbps, the upper limit of 3G is merely 3 Mbps.

In addition to that, usually, neither 3G nor 4G cover the entire surface of a country, which implies that the users have no other choice than to use 3G if 4G is not available at their current location, or even 2G. An interactive map published by the Belgian Institute for Postal services and Telecommunications (BIPT) [40] illustrates the coverage of 2G, 3G, and 4G in Belgium. As an example, Figure 3.3 shows the 4G coverage.

The performance also depends on the device itself. Indeed, some devices simply do not support 4G. The operating system, available RAM, and settings may also prevent the users from experiencing a satisfactory Internet connection.

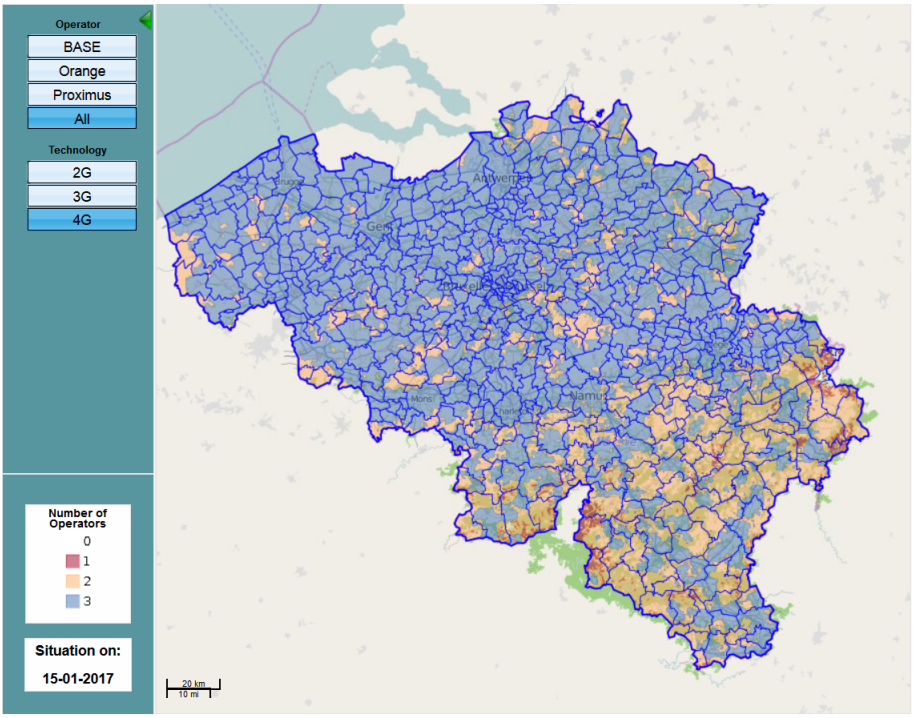


Figure 3.3: Coverage of the 4G technology in Belgium (courtesy of BIPT).

Chapter 4

DNS and Anycast

Summary. This chapter sheds light on the domain name system (DNS) and its use of anycast. In particular, we explain in Section 4.1 the basic principles behind DNS and why it is so crucial for the Internet to work in a smooth way. In Section 4.2, we detail the distributed architecture of the domain name system. Finally, we provide in Section 4.3 an overview of the three analysed DNS services, namely K-Root DNS, F-Root DNS, and Google DNS.

4.1 Overview

The main task of DNS is to provide name-to-IP mappings.

Each Internet host is identified by an IP address which can be used to contact the given host. However, IP addresses are a rather inconvenient way for users to interact with hosts, as they are not easy to remember. Instead, numerous hosts are assigned so-called *hostnames* that can be used to reach them. For instance, for accessing the Google search engine, a user prefers to simply type `google.com` instead of `216.58.213.142`. DNS can thus be considered as a huge directory that maps hostnames to their corresponding IP addresses.

4.2 Architecture

The architecture used for DNS is rather complex. One possibility would have been to consider one single DNS server storing all the different name-to-IP mappings. However, this would mean a single point of failure and a huge amount of traffic to handle for one single server.

This is why a distributed, hierarchical architecture has been adopted since the early days of the naming system, in 1987 [41]. More precisely, the DNS servers are subdivided into three categories which interact with each other:

- *root DNS servers*: there are 13 root DNS servers worldwide, labelled A through M, managed by 12 different operators having volunteered to host a root server. They make up the so-called *root zone*. Most of these services use the anycast paradigm to ensure reliability and security. B-Root started to rely on anycast in May 2017, according to a post on the official website of the DNS root servers [42]. On the same website, one can find a map depicting the location of the different servers running the root DNS service. As of May 2017, there are almost 700 root DNS servers distributed around the world, even though 20% of them are hosted in North America. In this thesis, we investigate anycast for two root servers, namely F-Root and K-Root. There is no hierarchy between these 13 root servers: all of them are equally important and behave identically in DNS terms. The only differences are their operators, their names, and their IP addresses [43].
- *top-level domain* (TLD) *DNS servers*: this kind of DNS server is responsible for top-level domains such as `.org`, `.com`, and `.net`, as well as for country top-level domains such as `.lu`, `.fr`, and `.be`. The TLD servers must be registered with the root servers so that they know the given TLD servers.

- *authoritative DNS servers*: every organisation with publicly accessible hosts such as Web and mail servers has to provide DNS mappings for them. These DNS records are stored on the organisations' authoritative servers. To notify their existence, authoritative servers must register with TLD servers with the help of a registrar.

Let's have a look at an example to understand how a DNS request is processed. Suppose a client needs the DNS mapping for `www.google.com`. First, the request is sent to one of the root servers, and the contacted server returns the IP address of the appropriate TLD server responsible for the top-level domain `.com`. To this end, root servers maintain a *root zone file* [44], which contains the corresponding TLD mappings and can therefore be considered as a DNS "phone book". Second, the client sends her request to the suggested TLD server, providing her with a list of IP addresses of the authoritative servers for `google.com`. Third and finally, the user sends her DNS-mapping request to one of the indicated authoritative servers and retrieves one of the IP addresses corresponding to the hostname `www.google.com`. Figure 4.1 depicts a part of the DNS architecture. This kind of architecture has also the advantage that updates are easier to perform, as every organisation can independently update the DNS entries stored on the corresponding authoritative servers.

This example is a so-called *iterative* query, as the client issues all the different requests herself. However, a DNS request can also be carried out through *recursive* queries. In the latter scenario, the DNS servers (instead of the client) contact the different servers of the hierarchy in order to get the requested mapping. The choice between the two operation modes is made by server configuration, as explained by Microsoft [45]. However, still according to Microsoft, recursive queries are usually avoided to prevent denial of service attacks (DoS).

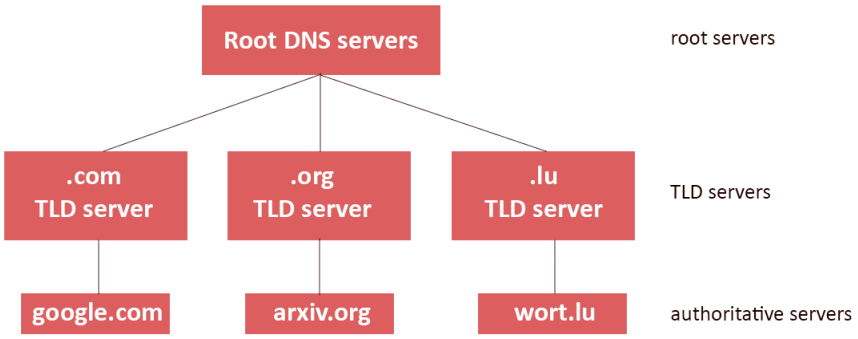


Figure 4.1: Part of the DNS hierarchy.

4.3 Analysed DNS Services

In this work, we investigate the anycast performance of three DNS services: F-Root DNS, K-Root DNS, and Google DNS. We selected these three services since they are both widely replicated services, with publicly available site locations and unicast IP addresses. This information allows to evaluate the performance of anycast routing relative to its "optimal" (in terms of unicast) site location.

The F-Root service is operated by the Internet Systems Consortium (ISC) and supports both IPv4 and IPv6. F-Root DNS includes more than 80 servers scattered across 39 different cities, covering 34 countries.

K-Root is a DNS service operated by the RIPE NCC (Network Coordination Centre). As F-Root, it runs both IPv4 and IPv6, but it is less well distributed. Indeed, K-Root DNS runs only 56 servers, which are distributed across 42 cities situated in 30 countries around the world.

Table 4.1 summarises the information we gathered about the locations of these root servers.

Continent	Number of F-Root servers	Number of K-Root servers
Europe	34	32
Asia	13	13
North America	25	8
South America	8	1
Oceania	2	1
Africa	7	1

Table 4.1: Distribution of the F- and K-Root servers at the continental level (as of August 2016).

Google DNS is a DNS resolution service provided by Google (and, contrary to F- and K-Root, not part of the root zone). The Google-DNS servers are scattered across 15 cities in 10 countries. However, they are very unevenly distributed; indeed, more than 40% of them are located in the US!

Chapter 5

Internet Diagnostic tools

Summary. In this chapter, we analyse the two most used Internet diagnostic tools, namely `ping` (Section 5.1) and `traceroute` (Section 5.2).

5.1 Ping

`Ping` [46] is a powerful tool written by Mike Muuss in 1983. Its purposes are twofold: first of all, it checks whether a host is reachable, and, if so, it measures the *round-trip time* (RTT) for messages sent from the source (i.e. the host on which `ping` is executed) to the target. To do so, `ping` relies on the Internet Control Message Protocol (ICMP) [47]: it sends *ICMP echo request* packets (*ping* packets) towards the target and waits for an *echo reply* (*pong* packets). This waiting time is then considered as the RTT. If the waiting time is higher than a given threshold, `ping` considers the target as unreachable. Moreover, `ping` provides additional information such as packet loss, the mean RTT, and errors that occurred during its execution. The general working scheme of this tool is depicted in Figure 5.1.

Besides the standard `ping`, there exist two variations of this diagnostic tool, relying on the principles of two different transport protocols [48]:

- **`UDP ping`:** to check whether a given target is alive, UDP `ping` sends UDP packets [49]. A target host is considered alive if the source receives an *ICMP port unreachable* message. Therefore, if possible, the source should send a packet to a port which is supposed to be closed.
- **`TCP ACK ping`:** TCP ACK `ping` works as follows: first of all, it sends a TCP ACK [50] to the target. The target is then reported as being alive if the source receives an RST message from the target.

In the study described in this thesis, we use `ICMP ping`, as it is implemented in the ALICE engine we are relying on for mobile measurements. This platform is presented in more detail in Chapter 6.

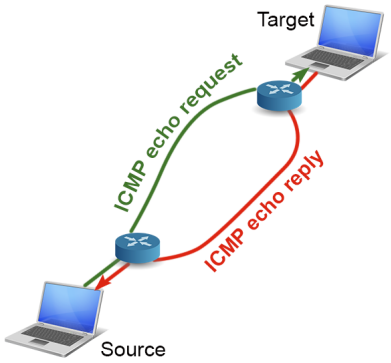


Figure 5.1: `Ping` overview.

5.2 Traceroute

`Traceroute` [51] is a tool allowing to infer a route from a source to any other target and was designed by Van Jacobson in 1989. `Traceroute` relies on UDP packets with a likely unused UDP destination port number (i.e. no process is listening on that port) whose destination address is the one of the target. This diagnostic tool makes use of the *Time To Live* (TTL) field and the IP forwarding mechanism in an ingenious way.

The basic idea behind `traceroute` is to discover the routers on the path by receiving *ICMP time exceeded* messages. `Traceroute` initially sends a UDP packet with a TTL equal to one and iteratively increments this value until the destination has been reached (or the TTL has exceeded a value defined by the user). The general working scheme of `traceroute` is depicted in Figure 5.2.

In a nutshell, when the *N*th packet arrives at the *N*th router, this router observes that the TTL has expired. According to forwarding rules of the IP protocol, this router discards the datagram, but sends an ICMP time exceeded message back to the source, which can then use the received packet to extract the IP address of the given router. The TTL of a datagram can thus be considered as its "life bar": each encountered router decreases it by one and, as soon as it is equal to zero, the packet "dies" (i.e. is not forwarded anymore). This TTL mechanism therefore prevents a cycling packet from wasting too many resources which could be useful for others. We have to point out that not every router sends an ICMP message back to the source. In this case, a timeout will be triggered and this hop will be indicated by a * instead of an IP address. When the target has been reached, the source receives an ICMP port unreachable message instead of a time exceeded message.

In addition, `traceroute` reports for each hop the RTT. Indeed, a timer is started when a new packet is sent, which allows `traceroute` to determine the time elapsed between the transmission of the UDP packet and the reception of the time exceeded/port unreachable message.

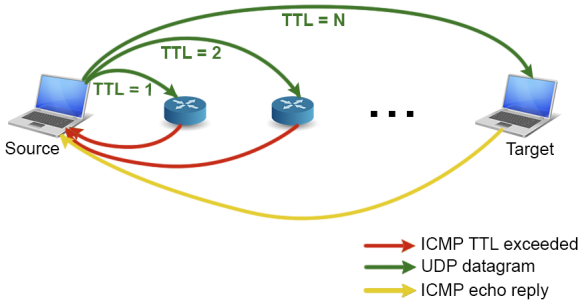


Figure 5.2: `Traceroute` overview.

As for `ping` (Section 5.1), there are two variations of the standard `traceroute` tool using two different transport protocols:

- **`ICMP traceroute`**: instead of sending UDP packets, `ICMP traceroute` is based on ICMP echo request messages. The target sends back an ICMP echo reply message when this tool is used. ALICE implements this version of `traceroute`.
- **`TCP traceroute`**: the disadvantage of `ICMP` and `UDP traceroute` is that firewalls and routers often block the ICMP protocol completely or disallow the ICMP echo requests (`ping` requests), and/or block various UDP ports. However, they rarely drop TCP packets on port 80 as this is the HTTP port. `TCP traceroute` sends TCP SYN packets, and the target responds with an RST packet if the port is closed and with a SYN/ACK packet otherwise.

The three variants of `traceroute` do not only distinguish themselves by the protocols they are relying on, but also by the information they can provide to their user. In their paper [52], Luckie et al. note that there are significant differences between the IP paths inferred by the three methodologies. Moreover, they observe that the three variants of `traceroute` are not on par when it comes to the discovery of IP links.

More precisely, the authors find that `UDP traceroute` infers the most unique links with respect to `ICMP` and `TCP traceroute`. Their results indeed suggest that the paths discovered by the ICMP and TCP methodologies are often the same. Concerning the number of discovered links, Luckie et al. observe that `TCP traceroute` derives by far the fewest links, that the UDP variant discovers the most links, but that `ICMP`

`traceroute` infers the most links that are not seen by any other method and reaches the most destinations. However, UDP `traceroute` does not really discover more links on the AS level, which suggests that the UDP variant provides more visibility on the intra-AS topology.

The authors assume that these differences are due to the fact that some routers use different forwarding policies depending on the transport protocol.

Chapter 6

Mobile Measurement Campaign

Summary. In this chapter, we present our measurement campaign. In particular, we introduce the mobile experiment engine ALICE we rely on in the context of this work (Section 6.1), the methodology we use to set up our measurements (Section 6.2), and provide an in-depth analysis of our dataset (Section 6.3).

6.1 ALICE – A Lightweight Interface for Controlled Experiments

In the context of our study, we launched and collected measurements from geographically distributed mobile vantage points through the ALICE engine [53, 54]. The characteristics making ALICE so appealing to use are the fact that it is designed as an Android library, which implies that it can be easily incorporated into existing applications, and the fact that it takes into account the resources of the mobile devices. More precisely, ALICE attempts to parallelise the requested measurements in order to minimise power consumption. Moreover, it imposes an upper limit to the network resources used by the experiments, so that the application does not use too much of the user's monthly data volume. As of December 2016, the ALICE engine has been run on over 2,100 unique mobile devices. Moreover, it has been integrated in three Android applications, namely NU Signals [55], Application Time [56], and Namehelp Mobile [57].

The current version of ALICE allows the user to launch the following types of active measurements and to gather useful information. The available active measurement probes are:

- DNS
- HTTP GET
- Ping
- Traceroute
- NDT (Network Diagnosis Tool – a network performance testing suite)
- IPerf (a bandwidth testing tool)

The user can retrieve the following information about the mobile device serving as a vantage point:

- the unique device identifier
- available sensors
- active and available network interfaces (3G, 4G, WiFi)
- traffic statistics (bytes sent/received by each application)
- the mobile client's anonymised current location (latitude/longitude coordinates)
- cellular signal strength
- available WiFi access points

ALICE operates both on the mobile client's device as well as on a centralised cloud service. The cloud service is responsible for distributing the experiment scripts to the concerned clients and makes sure that the requested measurements fit the capabilities of the mobile device. On the client side, ALICE schedules the measurements to be performed, executes the code, and retrieves the data.

6.2 Methodology

In the context of this study, geographically distributed mobile clients launch active measurements, more precisely `ping` and `traceroute` measurements, towards several unicast IPs and the corresponding anycast IPs of the three analysed DNS services to investigate whether the given clients get redirected to the optimal server in terms of latency and geographic distance.

Before launching any measurement, we preprocess the list of servers run by the concerned DNS services: for each service and each city hosting at least one server running the given service, we only keep one replica, as we assume that one server is representative for one city, and that measurements towards multiple servers in the same city would yield very similar results. In order to not overload the clients, we do not issue measurements towards each server of the three services. Instead, we choose for each client the five closest servers in terms of geographic distance. In the end, each client therefore issues 18 `ping` and `traceroute` measurements: towards five unicast IPs for each of the three services, and towards the corresponding anycast IPs. We refer to this set of measurements as *experiments*. The measurements are launched approximately once per hour.

We now clarify how we compute the geographic distance between a mobile client and a server. First, we determine the AS the client is hosted in thanks to her IP address. Whois data allows us to gather information about this AS. More precisely, we are interested in the country the AS is located in. In [58], Balakrishnan et al. argue that cell phone IP addresses are a very unreliable indicator for geographical locations. However, their accuracy is still reasonable at the country level. Finally, this information allows us to compute the geographic coordinates of the client: the ones representing the country her AS is hosted in. The approximate (latitude, longitude) coordinates of the unicast servers are known thanks to the airport situated in the city. After having computed the coordinates of both the client and the server, we calculate the distance between both using the classic Vincenty's formulae [59].

6.3 Dataset Presentation

We collected our data from September 2016 until April 2017, using the ALICE engine described in Section 6.1 and the methodology presented in Section 6. The analysis focuses on three major DNS services, namely F-Root, K-Root, and Google DNS. The data used for our study was retrieved from mobile devices while clients were connected to either WiFi or cellular networks. Our dataset can thus be divided into two sets: on the one hand, the set containing experiments launched from cellular networks (*CELL*); on the other hand, the set including the experiments issued from WiFi (*WIFI*). As measurements are collected opportunistically, based on connection availability and resource usage, the number of experiments launched over each connection type is not the same.

CELL Dataset CELL includes more than 20,000 experiments performed on cellular networks, issued from 151 different clients. Our cellular users are scattered across nearly 40 different countries, about 70% of the clients being located in the United States, Greece, Brazil, and France. Furthermore, the analysed clients are hosted in major ASes, such as AS 29247 (COSMOTE, Greece), AS 26599 (Vivo, Brazil), and AS 22394 (Verizon Wireless, United States). The experiments were launched using different network technologies: 91% were launched on 4G network, while 9% were issued with 3G.

WIFI Dataset WIFI encompasses thrice as many experiments than CELL, launched from 251 clients. The measurements were issued from nearly 50 different countries around the world. As for the CELL dataset, 70% of the WiFi clients are located in Greece and the United States. The most active clients (i.e. the ones having launched the most experiments) are hosted in ASes such as AS 6799 (Otenet, Greece), AS 9121 (TTNet, Turkey), and AS 7922 (Comcast, United States).

125 mobile users launched experiments from both cellular and WiFi networks.

The performed experiments are very unevenly distributed among the different clients. Figure 6.1 depicts this distribution. We can see that, for both cellular and WiFi, more than 90% of the clients have launched less than 500 experiments during the analysed seven months, i.e. less than three experiments per day on average.

One point which is worth mentioning is that a dataset involving cellular users is very difficult to gather. Indeed, a major drawback of active measurements is that they generate traffic on the client side. However, the clients' monthly data allowance is often rather limited and a lot of mobile users therefore do not want to dedicate a part of their data volume to this kind of study. We believe that, with these constraints, this amount of measurements is already a very good starting point.

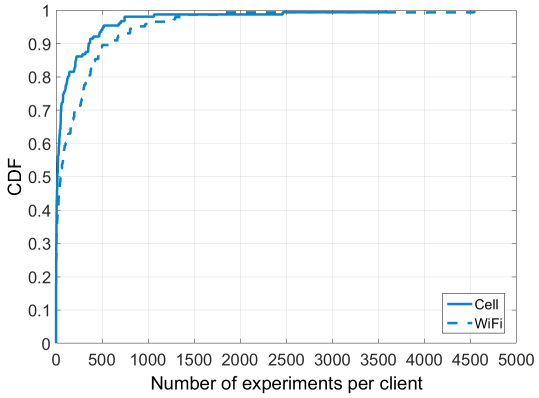
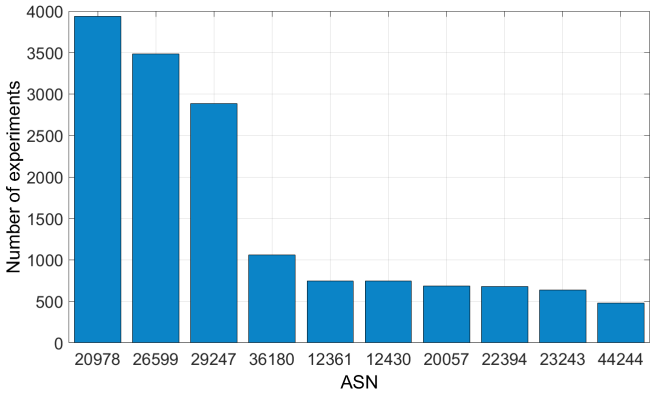


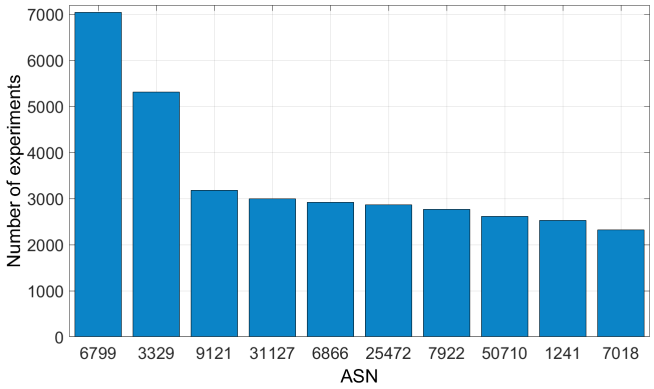
Figure 6.1: Distribution of experiments among the mobile clients.

Even though cellular clients issued experiments from 59 different ASes scattered around the world, only 19 of them present more than 100 experiments. We note the same for WIFI: 182 ASes are represented in this dataset, but the dataset includes less than 100 experiments for 139 of them. Figure 6.2 depicts the distribution of experiments among ASes for CELL (Figure 6.2a) and WIFI (Figure 6.2b). We observe that the different ASes cover several regions over the world: the top 10 ASes for CELL are located in countries such as Turkey, Brazil, Guatemala, Iran, the United States, and Greece, while the top 10 for WIFI sees the appearance of Slovakia, Cyprus, and Iraq.



(a) Experiment distribution for CELL.

Figure 6.2: Experiment distribution for both datasets at the AS level for top 10 ASes in terms of number of launched measurements.



(b) Experiment distribution for WIFI.

Figure 6.2: Experiment distribution for both datasets at the AS level for top 10 ASes in terms of number of launched measurements.

Chapter 7

Mobile Anycast Performance Analysis

Summary. In this chapter, we present our study of anycast performance in terms of latency with respect to unicast in two different ways. To this end, we introduce two metrics: the *relative latency increase* (Section 7.1), and the *absolute latency increase* (Section 7.2).

7.1 Performance Comparison with Relative Latency Increase

The first part of our DNS anycast study consists in assessing its performance in terms of percentages by comparing the RTTs towards the anycast servers and the corresponding five unicast replicas. To do so, we use the following formula, defined as *relative latency increase*:

$$\text{relative latency increase} = \frac{(\text{RTT}_{\text{anycast}} - \text{RTT}_{\text{optimal}}) \times 100}{\text{RTT}_{\text{optimal}}}$$

where $\text{RTT}_{\text{anycast}}$ refers to the RTT towards the examined anycast server and $\text{RTT}_{\text{optimal}}$ denotes the smallest observed RTT among the six `ping` measurements. Multiplying the fraction by 100 ensures that this increase is expressed in percentages. A relative latency increase of zero means that anycast is optimal in terms of latency; in other words, negative values cannot occur. Our metric is similar to the relative error, a metric very frequently used in the field of machine learning.

Figure 7.1 shows the relative latency increases we obtained for both CELL and WIFI. We can see that anycast performs well in about 60% of the measurements of our two datasets, i.e. anycast provides optimal latency with respect to the geographically closest unicast replicas. However, for the remaining 40%, anycast is suboptimal and the relative latency increase higher than 100% in approximately 10% of the carried out measurements.

We now investigate the performance of anycast for each of the three DNS services separately on cellular networks. The corresponding latency increases are presented in Figure 7.2. It demonstrates that all of the three services are confronted with anycast performance issues. Nevertheless, we can clearly note that the performance of anycast seems to be the worst for K-Root: while the latency towards the anycast server is optimal in about 70% of the measurements for F-Root and Google DNS, this is the case for less than 40% when it comes to K-Root.

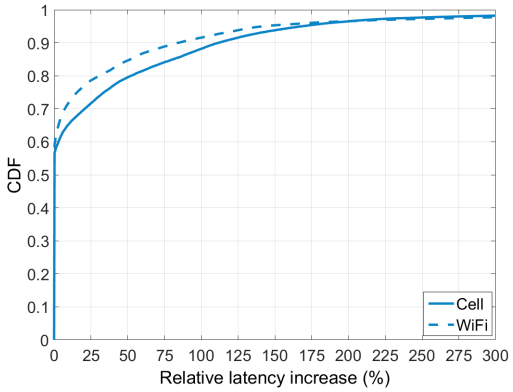


Figure 7.1: Relative latency increases observed for CELL and WIFI.

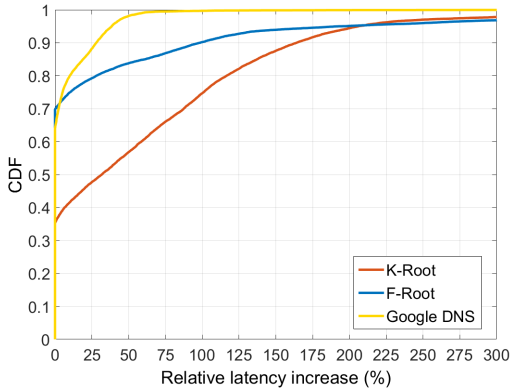


Figure 7.2: Relative latency increases observed for the three DNS services on cell.

Next, we analyse whether this performance problem occurs only in specific regions or if it happens to be more general. Figures 7.4a to 7.4f depict the relative latency increases obtained by cellular clients in each encountered country in case anycast is suboptimal. The six figures show that there is no particular pattern at the continental level. Nevertheless, several points are worth mentioning. In Europe (Figure 7.4a), 7 out of 13 countries present relative latency increases of at least 200%, while anycast performs reasonably well in the five remaining ones. We observe the same kind of behaviour for anycast in Asia (Figure 7.4b). In North America (Figure 7.4c), the performance of the anycast paradigm with respect to unicast is rather poor: for *all* of the examined countries, the relative latency increase exceeds 1000%! The three remaining continents, i.e. South America (Figure 7.4d), Oceania (Figure 7.4e), and Africa (Figure 7.4f), are unfortunately not very well covered by our CELL dataset. Nevertheless, we see that the relative latency increases provided by Australian cellular clients are relatively low, with the maximum being less than 300%. Our measurements suggest that anycast performs surprisingly well in Africa, but the number of measurements for this region is extremely low. Table 7.1 summarises additional statistics we gathered about the relative latency increases in case the performance of anycast was suboptimal, and underlines the heterogeneous behaviour of this paradigm across the encountered countries.

Additionally, we analyse the relative latency increases observed in the top 10 ASes in terms of number of launched measurements on cellular networks (the list of these ASes as well as the corresponding number of measurements can be found in Chapter 6). The obtained relative latency increases are plotted in Figure 7.3. As for the regional analysis, we cannot infer from our results that anycast performance in some ASes is significantly different than the one in other ones. Of course, we do note some differences, but, in our opinion, they are not striking enough to conclude that suboptimal anycast performance is an AS-specific problem.

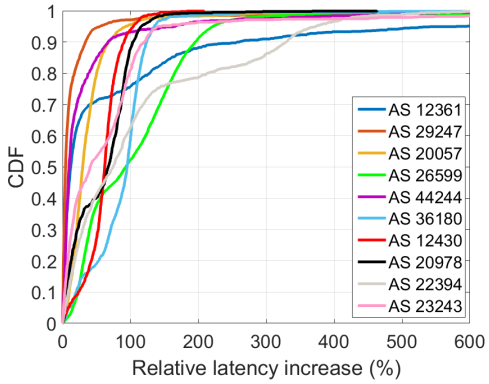
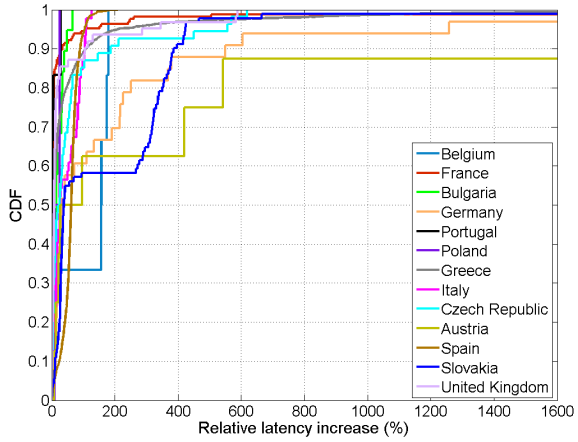
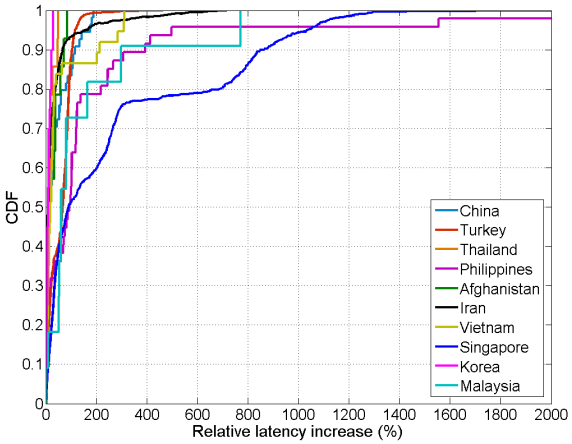


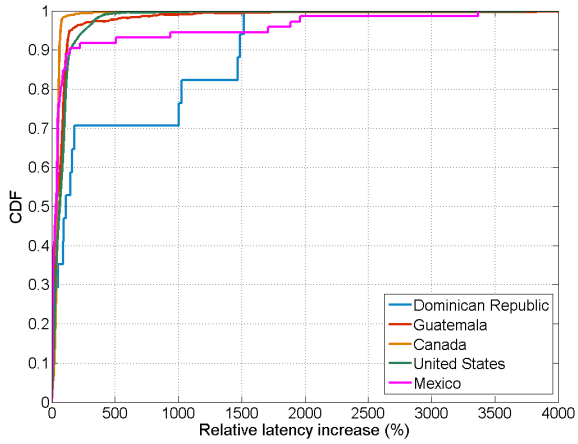
Figure 7.3: Relative latency increases in case anycast is suboptimal – AS level (top 10 ASes in terms of number of launched measurements in cellular networks).



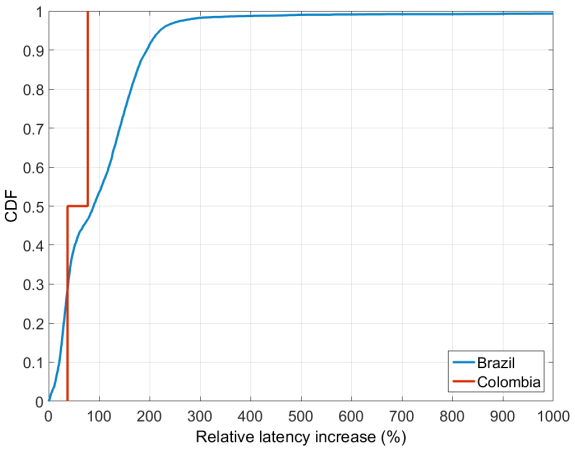
(a) Relative latency increases in Europe.



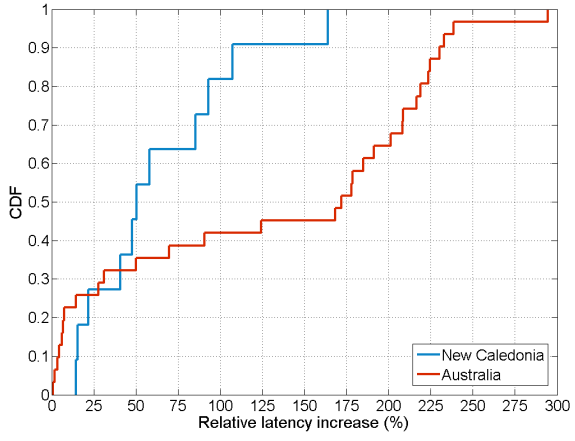
(b) Relative latency increases in Asia.



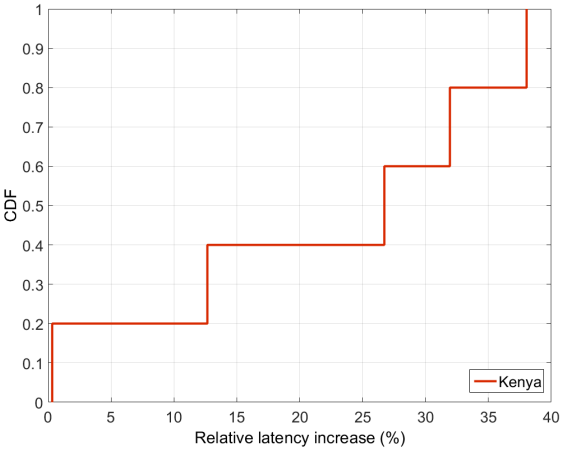
(c) Relative latency increases in North America.



(d) Relative latency increases in South America.



(e) Relative latency increases in Oceania.



(f) Relative latency increases in Africa.

Figure 7.4: Relative latency increases in case anycast is suboptimal – continental level.

Continent	Country	% anycast suboptimal	total # measurements	Mean	Median	Standard deviation	10% percentile	25% percentile	75% percentile	90% percentile
Europe	Belgium	50%	12	115.0%	157.0%	74.3%	10.6%	48.1%	170.4%	177.6%
	France	39.1%	422	422.8%	2.6%	3793.3%	0.5%	1.1%	5.6%	31.6%
	Bulgaria	42.2%	45	20.1%	13.3%	17.6%	1.6%	7.4%	32.2%	41.9%
	Germany	2.6%	1288	202.2%	30.2%	366.0%	11.7%	17.2%	218.0%	515.0%
	Portugal	33.3%	18	6.8%	2.4%	10.0%	0.6%	1.1%	5.1%	17.4%
	Poland	66.7%	3	17.0%	17.0%	7.9%	10.6%	13.0%	21.0%	23.4%
	Greece	33.3%	11,996	75.1%	7.7%	418.5%	0.9%	2.7%	31.0%	107.6%
	Italy	80.7%	57	45.3%	27.3%	39.5%	1.9%	13.6%	83.2%	104.7%
	Czech Republic	58.1%	93	72.8%	22.4%	142.5%	2.4%	4.7%	57.4%	174.4%
	Austria	0.9%	847	419.6%	62.7%	710.0%	11.3%	19.6%	450.3%	1047.7%
	Spain	37.7%	2220	60.4%	62.0%	26.7%	23.2%	48.4%	73.2%	90.4%
	Slovakia	53.8%	169	260.9%	38.4%	939.2%	11.4%	27.2%	328.1%	387.1%
	United Kingdom	21.8%	284	43.8%	8.4%	115.7%	1.4%	4.1%	12.3%	104.8%
Asia	China	50%	108	41.6%	13.9%	50.2%	6.6%	10.0%	55.2%	117.9%
	Turkey	91.8%	942	59.7%	67.2%	44.6%	6.2%	16.9%	89.1%	104.4%
	Thailand	11.9%	118	18.2%	14.4%	14.0%	6.7%	7.6%	18.2%	42.7%
	Philippines	55.3%	85	263.5%	95.0%	885.7%	3.6%	9.0%	124.1%	341.6%
	Afghanistan	82.4%	17	27.7%	11.9%	26.6%	5.7%	6.0%	38.2%	67.8%
	Iran	57.0%	1443	34.9%	9.6%%	82.5%	2.2%	4.6%	28.2%	66.6%
	Vietnam	56.1%	66	51.6%	17.9%	86.8%	4.8%	8.1%	32.1%	207.1%
	Singapore	77.1%	1042	274.6%	89.1%	352.0%	11.7%	30.7%	292.6%	861.8%
	Korea	74.1%	27	11.9%	9.5%	8.5%	3.1%	5.7%	17.9%	23.7%
	Malaysia	8.2%	134	148.5%	62.1%	211.8%	9.8%	52.6%	122.7%	299.0%
North America	Dominican Republic	2.9%	588	434.8%	113.8%	576.0%	8.3%	15.6%	1005.3%	1477.1%
	Guatemala	53%	1732	87.2%	40.4%	337.8%	3.3%	10.2%	91.7%	120.0%
	Canada	36.4%	1332	45.6%	43.4%	37.1%	20.2%	32.7%	52.9%	61.6%
	United States	56.0%	7852	90.4%	64.1%	280.6%	11.8%	25.7%	105.2%	146.4%
	Mexico	54.1%	135	175.4%	34.6%	533.4%	3.0%	7.3%	53.6%	139.5%
South America	Brazil	52.7%	10,773	116.5%	89.0%	254.9%	21.1%	34.1%	151.7%	194.5%
	Colombia	50%	4	57.4%	57.4%	20.0%	41.4%	47.4%	67.5%	73.5%
Australia	New Caledonia	0.8%	1334	63.4%	50.3%	43.4%	15.4%	31.2%	89.0%	107.3%
	Australia	37.8%	82	129.3%	171.8%	95.0%	4.3%	21.0%	212.6%	230.1%
Africa	Kenya	33.3%	15	21.9%	26.7%	13.6%	5.2%	12.6%	31.9%	35.5%

Table 7.1: Statistics concerning relative latency increases in case anycast is suboptimal – country level.

7.2 Performance Comparison with Absolute Latency Increase

The relative latency increase metric has a disadvantage. Indeed, it does not reveal any information about the perceived anycast latency: for instance, a relative latency increase of 100% is obtained when the anycast latency is 4 ms and the optimal unicast latency is equal to 2 ms ($\frac{4-2}{2} \times 100 = 100\%$). However, our metric has the same value when the anycast latency is equal to 200 ms and the optimal unicast one to 100 ms. In the first case, anycast is indeed suboptimal, but the impact on latency is relatively negligible for the mobile clients.

Therefore, we now introduce the *absolute latency increase* metric, simply defined as:

$$\text{absolute latency increase} = \text{RTT}_{\text{anycast}} - \text{RTT}_{\text{optimal}}$$

where, as for the relative latency increase, $\text{RTT}_{\text{anycast}}$ refers to the RTT towards the examined anycast server and $\text{RTT}_{\text{optimal}}$ denotes the smallest observed RTT among the six `ping` measurements; again, this implies that negative values cannot occur.

We compute the absolute latency increases in case anycast is suboptimal for one country of each continent. More precisely, we choose the country for which we observe the worst relative latency increases. One exception is Europe, for which we select Slovakia instead of Austria, as we have significantly more measurements for this country. Figure 7.5 illustrates our results. Even though most of the absolute latency increases are below 500 ms (which is already a very high value for a RTT!), besides for the Dominican Republic, we observe a high amount of absolute latency increases which are higher or equal to 100 ms. To grasp even better what this means, we can note that a packet theoretically travels at $2/3$ of the speed of light (which corresponds to the maximum speed in optical fibre [60]), i.e. at approximately 200,000 kilometres per second – ignoring the processing delay induced by the routers on the path. This means that a mobile client's probe having yielded an RTT higher or equal to 100 ms might have travelled at least 20,000 kilometres, corresponding to a return trip between San Francisco and Berlin!

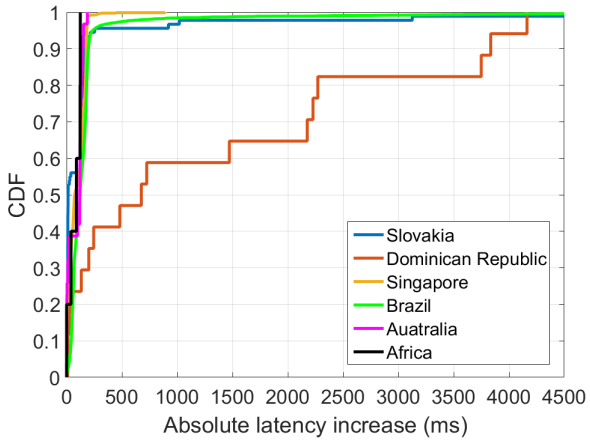


Figure 7.5: Absolute latency increases in case anycast is suboptimal in one country per continent.

Chapter 8

Mobile Anycast Distance Analysis

Summary. In this chapter, we investigate the geographic distances between mobile clients and the DNS replicas. In Section 8.1, we present the *Out-Of-Country Problem*, i.e. the phenomenon that clients' packets get routed outside their home country in spite of local alternatives. In Section 8.2, we analyse the distances travelled by packets in more general terms and observe that the journey of packets is frequently longer than necessary, with a negative impact on latency. We refer to this problem as the *Travel Distance Problem*. In Section 8.3, we begin to explore the root causes of the encountered problems and find three classes of mobile anycast anomalies.

Remark. We leave out the Google DNS service from the analyses presented in this chapter. The reason for this is that this part of the study heavily relies on geographical locations of replicas and that Google DNS servers are extremely difficult to geolocate. Indeed, nearly all of the servers are geolocated in the US, while we know thanks to official information that a large number of Google DNS servers are hosted in other regions.

8.1 The Out-Of-Country Problem

Our findings presented in Chapter 7 suggest that anycast is suboptimal in terms of latency for a non negligible fraction of measurements. Further investigation shows that mobile clients get often routed towards a suboptimal anycast replica in terms of geographical distance, which negatively impacts the perceived performance. One could argue that the most impacting factor on latency in wireless networks is the signal strength and that geographic distance is not very influential. However, Figure 8.1 clearly shows that high geographical distance is correlated with large latency, underlining that the distance between cellular clients and anycast servers is an important aspect to take into account when it comes to performance optimisation. We observe a similar behaviour for WiFi clients.

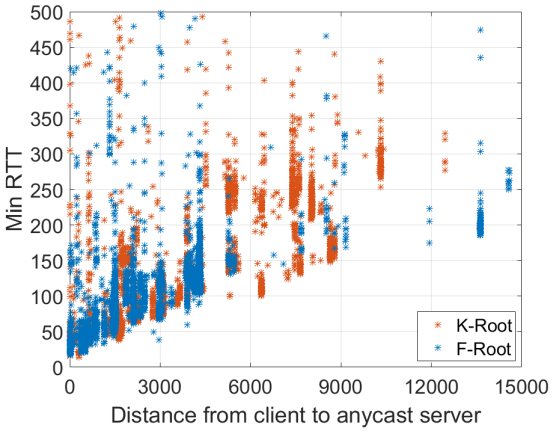


Figure 8.1: Potential correlation between geographical distance and minimum RTT for cellular measurements.

This leads us to the analysis of the geolocation mapping between mobile clients and anycast replicas. Our initial results show that cellular clients are in general frequently routed towards a replica hosted outside the country they are residing in, despite local alternatives. To characterise distances between clients and anycast replicas, given that anycast IP addresses cannot be geolocated [10], we geolocate the penultimate hop of the `traceroute` path connecting the mobile client to the anycast server to estimate its location. Our mobile clients recorded their geographic location, anonymised to a 10 km² area, through the ALICE engine.

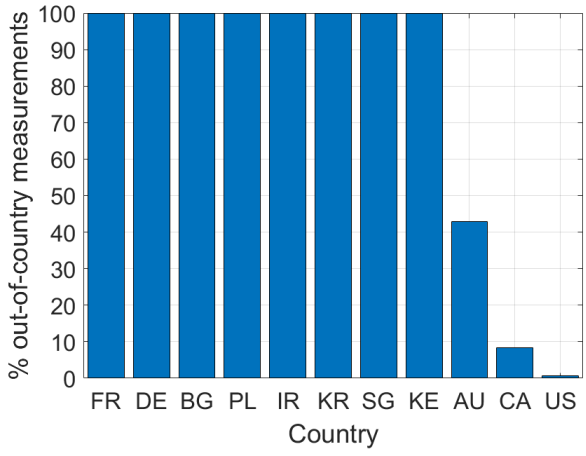
Tables 8.1 and 8.2 summarise the collected statistics for cellular clients. We can point out that the sent packets happen to be redirected to locations lying very far away from their country of origin. For instance, we observe packets from the United States being sent to K-Root replicas in Iran and United Kingdom! We see the same abnormal behaviour for F-Root DNS, for example with cellular users in Germany being routed to the US.

However, we need to be aware that geolocation databases are not 100% reliable [61]. The statistics provided in this thesis might therefore contain some inaccurate information, due to a wrong IP-to-location mapping. In the context of this study, we rely on the IP2Location DB5.LITE database [62].

Client's country	Contacted server's location	Any local replica?
Dominican Republic	United States	No
France	France	Yes
Bulgaria	Bulgaria	Yes
Belgium	Iran	No
Germany	**United Kingdom**	**Yes**
Brazil	United Kingdom, United States, Germany	No
Turkey	Iran, Germany	No
Poland	**Germany**	**Yes**
Guatemala	United States	No
Greece	**Greece, Germany**	**Yes**
Canada	**United States, United Kingdom**	**Yes**
Iran	**Iran, Germany, India,**	**Yes**
Czech Republic	**Czech Republic, Germany**	**Yes**
Australia	Australia	Yes
United Kingdom	United Kingdom	Yes
Spain	**United Kingdom, Russia, Latvia**	**Yes**
Kenya	Johannesburg	No
United States	**United States, United Kingdom, Iran**	**Yes**
Slovakia	Germany	No
Korea	United States, United Kingdom	No
Singapore	Japan, India	No
Mexico	United States	No

Table 8.1: Location of contacted K-Root anycast replicas for cellular clients.

(a) K-Root DNS.



(b) F-Root DNS.

Figure 8.2: Frequency of out-of-country travelling for cellular clients to reach K- and F-Root.

Client's country	Contacted server's location	Any local replica?
Dominican Republic	United States	No
France	**Netherlands**	**Yes**
Bulgaria	**Netherlands**	**Yes**
Belgium	Netherlands	No
Germany	**United States, Netherlands**	**Yes**
Brazil	Brazil	Yes
Turkey	United States	No
Poland	**Czech Republic**	**Yes**
Guatemala	United States, Canada	No
Greece	United States, Netherlands, Germany	No

Client's country	Contacted server's location	Any local replica?
Canada	**Canada, United States**	**Yes**
Iran	**United Arab Emirates, Netherlands**	**Yes**
Czech Republic	Czech Republic	Yes
Australia	**Australia, United States**	**Yes**
Spain	Spain	Yes
Kenya	**Netherlands**	**Yes**
United States	**United States, Singapore**	**Yes**
Slovakia	United States	No
Korea	**United States**	**Yes**
Singapore	**United States**	**Yes**

Table 8.2: Location of contacted F-Root anycast replicas for cellular clients.

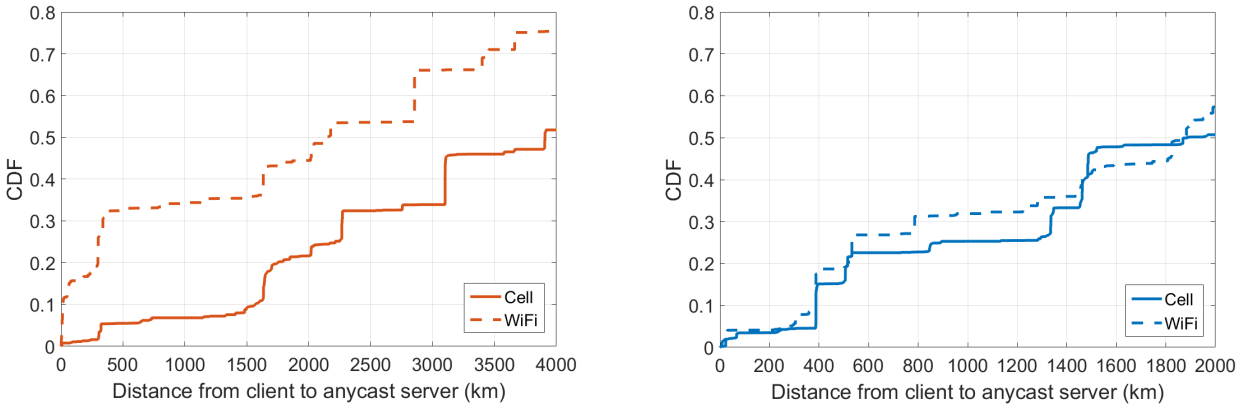
We analyse how frequently this out-of-country phenomenon occurs in the affected countries. Figures 8.2a and 8.2b depict our results. Our evaluation reveals that the packets travel abroad very often: for K-Root, cellular clients in 50% of the concerned countries are systematically redirected abroad, while this is the case for even more than 70% of the concerned countries when looking at F-Root. Moreover, the out-of-country observation occurs in fewer countries for K-Root than for F-Root. A striking fact is that cellular clients residing in the US are routed towards foreign K-Root replicas more than 70% of the time, while this holds for only merely 1% of the measurements when probing F-Root.

8.2 The Travel Distance Problem

The out-of-country analysis detailed in Section 8.1 already gives a good intuition about the journey of the issued packets. However, it could be the case that clients are located at the border of a country and that it would be cheaper in terms of distance to contact a unicast server hosted in the neighbour country instead of

letting their request traverse the whole country to reach a local alternative. This part of the study focuses on the analysis of distances between mobile clients and their assigned anycast DNS servers.

Figure 8.3a presents the travel distance between clients and anycast servers ($D_{\text{client}\rightarrow\text{anycast}}$) for K-Root DNS. We can easily see that there is a significant difference between cell and WiFi: $D_{\text{client}\rightarrow\text{anycast}}$ is smaller than 4,000 kilometres for approximately 75% of the experiments carried out on a WiFi network, whereas this holds for merely 50% of the experiments issued from cellular clients. While we can observe the same phenomenon for F-Root in Figure 8.3b, the differences are not as significant as for the K-Root service. Moreover, the distances plotted in Figure 8.3 suggest that the mobile clients are in general located closer to F-Root replicas than to K-Root ones. This can be explained by the number and geographical distribution of the different servers, presented in Chapter 4.



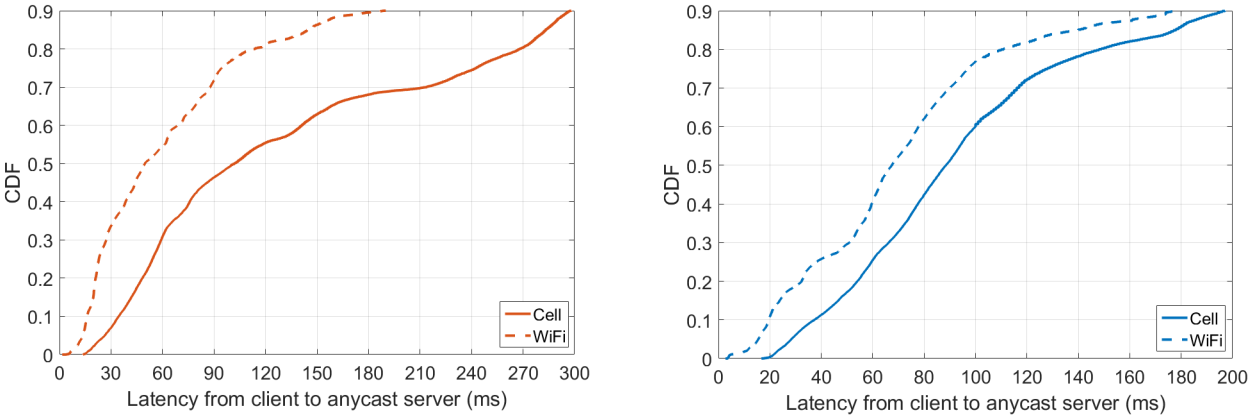
(a) Distance to K-Root servers.

(b) Distance to F-Root servers.

Figure 8.3: Travel distance from clients to the assigned anycast servers.

8.2.1 Impact on latency

As already stated in Section 8.2, the additional distance travelled does impact users' performance. We evaluate the impact of the geographical distance between the clients and the replicas in terms of latency. The results are depicted in Figure 8.4. As expected, the recorded latencies are higher for the measurements carried out on cellular networks than for the ones performed on WiFi. Again, the difference is more pronounced for K-Root, as we can conclude from Figure 8.4a: 90% of the WiFi experiments yielded a RTT lower than 180 ms, versus only 70% on cell. A comparison of Figures 8.4a and 8.4b shows that the distribution of latencies is similar for K-Root and F-Root on WiFi. However, we obtained worse performance for K-Root compared to F-Root on cellular networks: 80% of the experiments output a RTT lower than 150 ms for F-Root, but less than 65% for K-Root DNS. Note that Cisco considers 150 ms as being the upper delay limit to get an acceptable quality for most voice applications [63].



(a) RTTs for K-Root servers.

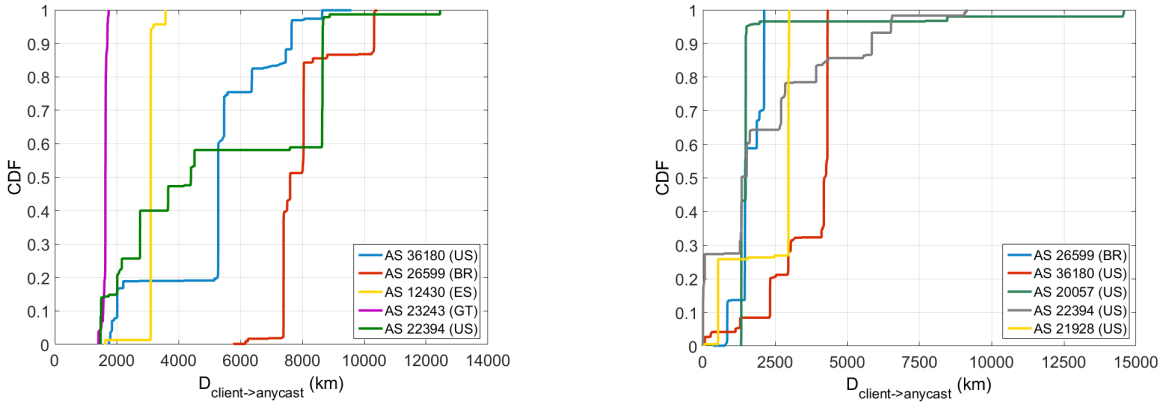
(b) RTTs for F-Root servers.

Figure 8.4: Round-trip time from clients to the assigned anycast servers.

8.2.2 Where does the Travel Distance Problem occur?

The observed phenomenon is not limited to specific geographical regions or a few particularly mismanaged ASes, but appears to be more general. We focus this part of our analysis on the five ASes which provide the most measurements for the two DNS services with clients in cellular networks. This leaves us with 2,310 measurements issued from 16 users for K-Root and 1,860 measurements retrieved from 15 mobile clients for F-Root DNS. Figure 8.5 presents our results.

A first noteworthy point is that F-Root DNS replicas seem to be closer to the clients, regardless of the AS. Indeed, for 80% of the launched measurements towards F-Root anycast servers, $D_{\text{client}\rightarrow\text{anycast}}$ is less than approximately 4,300 kilometres, while this distance is larger than 8,500 kilometres for 20% of the measurements issued towards K-Root servers! From this figure, we can distinguish two sets of ASes: those which present distances staying rather stable (for example AS 12430 – Vodafone, Spain), and those for which we observe significant variability on distances travelled by their users (a good example is AS 22394 – Verizon Wireless, United States).



(a) $D_{\text{client}\rightarrow\text{anycast}}$ observed for K-Root.

(b) $D_{\text{client}\rightarrow\text{anycast}}$ observed for F-Root.

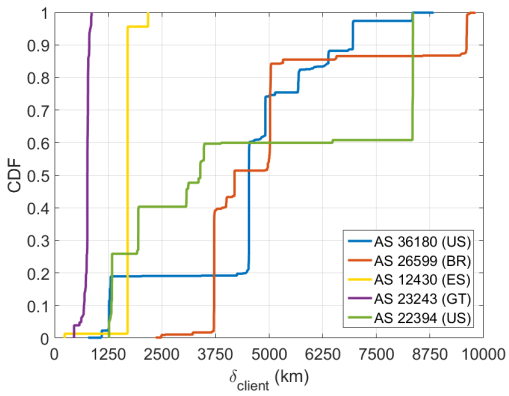
Figure 8.5: Distances between the cellular clients and their assigned anycast servers in top 5 ASes in terms of number of measurements.

A possible explanation for the long distances would be that our cellular clients are simply far away from all the available replicas. However, our investigations show that this is not necessarily the case. For this analysis, we compute:

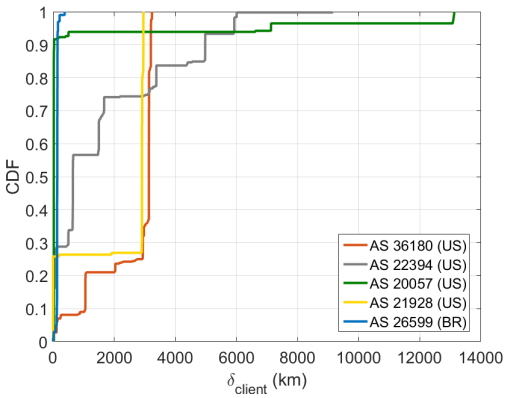
1. the distance between the cellular client and her assigned anycast server, $D_{\text{client}\rightarrow\text{anycast}}$;
2. the distance between the client and the geographically closest unicast replica, $D_{\text{client}\rightarrow\text{unicast}}$;

3. the difference $D_{\text{client}\to\text{anycast}} - D_{\text{client}\to\text{unicast}}$, denoted δ_{client}.

Figure 8.6 presents δ_{client} for the measurements launched from the five ASes providing the most experiments for K- and F-Root DNS. We can easily infer from this graph that the clients are most of the time routed to a suboptimal replica and that this issue does not seem to be specific to one AS or region. Nevertheless, comparing Figures 8.6a and 8.6b leads us to the assumption that the raised mapping problem is significantly worse for K-Root than for F-Root, even though the mappings for F-Root are far from optimal. Clear examples for the K-Root phenomenon are clients residing in the US, but being routed towards anycast servers located in London (UK) and Tehran (Iran)! This is even more surprising when considering the fact that seven K-Root servers are widely distributed across the United States: two are hosted on the East Coast, four on the West Coast, and one in Central America (as of May 2017). Furthermore, we observe that, in each of the examined ASes, some cellular clients seem to be redirected to the nearest anycast F-Root server (δ_{client} very close to zero), while the ideal case never occurs for the K-Root service.



(a) δ_{client} observed for K-Root.



(b) δ_{client} observed for F-Root.

Figure 8.6: Geographical distance differences between cellular client → anycast server and client → closest unicast replica in top 5 ASes in terms of number of measurements.

8.3 Why travelling so far?

In this section, we begin to explore some of the possible causes of anycast anomalies for cellular clients. We first investigate the AS-path lengths between anycast and nearest unicast paths, finding that path length does not seem to be correlated with distant anycast servers. We then look beyond AS-path length, finding three classes of anomalies specifically related to, or common to our cellular clients. These include:

1. distant client packet gateways;
2. poor anycast routing within Tier-1 networks;
3. improper routing *within* cellular networks.

8.3.1 AS-path analysis

We analyse the lengths of the AS paths leading from the mobile clients to their assigned anycast server and to the geographically closest unicast replica. In particular, we focus on the measurements with δ_{client} higher than 1,000 kilometres, which corresponds to more than 80% of the experiments for K-Root and to 70% of them for F-Root. Figure 8.7 presents the obtained AS-path length differences. A negative value indicates that the AS path towards unicast was longer than the one towards anycast. The graph suggests that the anycast servers are still often closer to the mobile clients in terms of traversed ASes, even though they are geographically significantly further away from the users: the paths client → anycast server are shorter than the paths client → unicast replica in nearly 40% of the measurements towards K-Root and in more than 75% towards F-Root.

This can be explained by the way anycast operates [1]: the server the client is assigned to is selected based on BGP's best-path notions such as shortest AS-path length, Hot Potato criterion, etc., and it is therefore not surprising to see that the paths leading to the anycast replicas are shorter. Figure 8.8 underlines our statement

regarding path lengths: while more than 70% of the paths to F-Root anycast servers have a length of at most six AS hops (more than 60% if we consider K-Root), 60% of unicast paths for the F-Root DNS service are longer than six hops (50% when looking at K-Root).

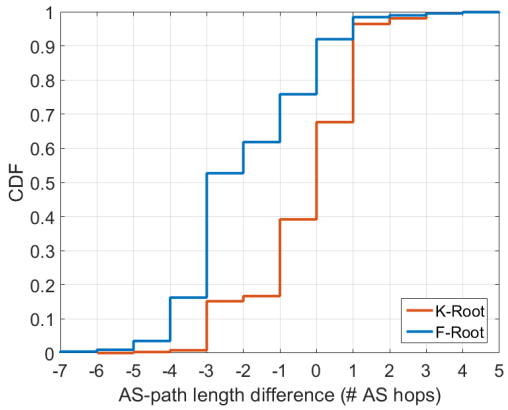


Figure 8.7: Path-length difference on the AS level between the paths client → anycast server and client → closest unicast replica on cell.

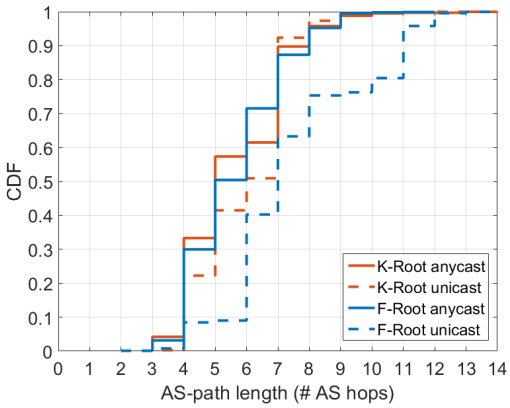


Figure 8.8: AS-path lengths observed for both anycast and unicast on cell.

8.3.2 Cellular anomaly classification

Further investigation into anomalous anycast routing for our cellular clients revealed three main classes of anycast performance problems. While by no means representative of all problems encountered by cellular clients, these common anycast routing problems are either unique, or more common, for cellular rather than fixed-line clients.

Distant packet gateway

As has previously been reported [64], a cellular client's packet gateway (PGW) largely determines her network position and locality, since all client traffic is routed through that particular PGW. This implies that client packets first encounter Internet routing once they are beyond the PGW. In certain instances, cellular clients can be assigned to distant PGWs. We find that knowing the location of a client's PGW can greatly aid in diagnosing anomalous anycast routing in these cases. An impressive example is shown in Table 8.3a. These paths correspond to a client located in Boston (MA), but whose PGW is situated in San Jose (CA). While the closest unicast replica is hosted in St. George (UT), the client's anycast request was routed to London. In this scenario, the journey of the packet is not only costly in terms of kilometres because of the suboptimal geolocation mapping between the client and an anycast server, but also due to the fact that the packet needs to travel more than 5,000 kilometres to reach the PGW!

Tier-1 routing

We observe that many of the instances of poor anycast performance occur when paths traverse Tier-1 transit networks. While this effect has been previously reported for fixed-line networks [65], our results confirm this phenomenon. We find that this problem is more pronounced for cellular networks, since the paths employed by packets sent from cellular networks traverse Tier-1 ASes more frequently than packets sent from fixed-line broadband [54]. The problem we observe with large Tier-1 networks is that they often route clients to the same anycast replica regardless of where clients enter their network. Many problem cases we investigate appear to be caused by packets remaining in the transit network until routed to a distant destination. We detect this behaviour in Tier-1 networks with varying levels of consistency. For example, we find clients entering AT&T (AS 7018) split anycast destinations between sites in Reno (NV) and London. We also note that clients routed through Abovenet (AS 6461) are consistently routed to K-Root sites in London. One of these cases is illustrated in Table 8.3b. This example corresponds to a cellular client residing in Los Angeles and whose PGW is in the same city. Even though the geographically closest unicast server is in Reno (NV), her packet is routed to a

server situated in London. Analysing the corresponding `traceroutes` reveals that more than 50% of the `traceroute` hops in the path leading to the anycast server lie in Abovenet, a well-known Tier-1 AS. As opposed to the anycast path, the path connecting the cellular user to her nearest unicast server exits the AS 6461 fairly quickly: less than 25% of the hops are in this Tier-1 AS.

Improper cellular network routing

We find cases where paths leaving towards the anycast and nearest unicast servers diverge immediately after exiting a client's PGW. As illustrated in Table 8.3c, the paths diverge right after the they exit the client's PGW. Interestingly, these diverging paths remain in the same AS (275699 – Vivo, Brazil), and even have the same next hop AS. While we cannot know what caused this exact instance, we observe for several operators multiple transit providers connected to or very near cellular network PGWs. As we noticed in the previous class, the choice of transit can play a large role in the behaviour of anycast routes. This is especially true if the provider is a Tier-1 network, as many commonly are for large cellular networks.

Client → anycast server			Client → unicast server		
Hop	**IP**	**ASN**	**Hop**	**IP**	**ASN**
1	10.96.0.3	-	1	10.96.0.3	-
2	100.127.126.35	-	2	100.127.126.36	-
3	192.168.189.5	-	3	192.168.189.5	-
4	192.168.125.2	-	4	192.168.125.2	-
5	192.168.126.12	-	5	192.168.126.12	-
6	208.185.160.182	6461	6	208.185.160.182	6461
7	208.185.160.181	6461	7	208.185.160.181	6461
8	64.125.28.41	6461	8	64.125.28.45	6461
9	64.125.30.230	6461	9	64.125.31.15	6461
10	64.125.31.218	6461	10	63.146.26.253	209
11	64.125.29.18	6461	12	63.234.254.142	209
12	64.125.29.208	6461	15	209.33.214.250	11071
13	64.125.29.127	6461			
14	64.125.30.237	6461	**end-to-end RTT: 160 ms**		
15	64.125.31.193	6461			
16	64.125.27.50	6461			
17	213.161.79.50	6461			
18	193.0.14.129	25152			
end-to-end RTT: 258 ms					

(a) Distant client packet gateway scenario.

Client → anycast server			Client → unicast server		
Hop	**IP**	**ASN**	**Hop**	**IP**	**ASN**
1	255.0.0.0	-	1	255.0.0.0	-
3	255.0.0.1	-	3	255.0.0.1	-
5	10.170.208.11	-	5	10.170.208.11	-
6	10.164.162.210	-	6	10.164.162.210	-
7	10.164.165.3	-	7	10.164.165.3	-
8	208.185.160.101	6461	8	208.185.160.101	6461
9	64.125.30.72	6461	9	4.125.30.72	6461
10	64.125.30.184	6461	10	64.125.26.5	6461
11	64.125.29.52	6461	11	64.125.12.154	6461
12	64.125.28.98	6461	12	12.122.129.234	7018
13	64.125.29.48	6461	13	12.122.31.134	7018
14	64.125.29.130	6461	14	12.122.18.13	7018
15	64.125.30.235	6461	15	12.122.160.29	7018
17	213.161.79.50	6461	16	12.116.94.238	7018
18	193.0.14.129	25152	17	208.79.242.141	11170
			18	208.79.242.186	11170
end-to-end RTT: 153 ms			19	74.118.156.91	11170
			end-to-end RTT: 58 ms		

(b) Tier-1 routing issue.

Client → anycast server			Client → unicast server		
Hop	**IP**	**ASN**	**Hop**	**IP**	**ASN**
3	177.79.213.38	26599	3	177.79.213.38	26599
4	187.100.51.69	27699	4	187.100.86.164	27699
5	187.100.193.254	27699	5	187.100.49.25	27699
6	187.100.192.246	27699	6	187.100.49.17	27699
7	213.140.39.70	12956	7	5.53.0.189	12956
8	176.52.250.246	12956	8	94.142.97.9	12956
9	195.22.199.89	6762	9	5.53.3.43	12956
10	195.22.209.107	6762	10	94.142.119.168	12956
11	195.66.224.183	8881	11	94.142.127.69	12956
12	193.0.14.129	25152	12	213.140.35.83	12956
			13	176.52.248.199	12956
end-to-end RTT: 271 ms			14	5.53.1.149	12956
			19	200.40.98.29	6057
			20	200.40.98.27	6057
			21	200.7.84.36	28000
			22	179.0.156.11	28000
			end-to-end RTT: 119 ms		

(c) Improper routing out of cellular network.

Table 8.3: Three classes of anycast anomalies encountered by our cellular clients.

Chapter 9

IP Assignment Dynamics

Summary. In this chapter, we perform a short analysis about the IP assignment dynamics, i.e. the frequency at which a different IP is assigned to the mobile client. In Section 9.1, we describe the IP distribution among clients, i.e. how many different IPs one gets assigned to and how frequently one changes from one IP to another. In Section 9.2, we assess the impact of IP assignment dynamics on DNS anycast performance: intuitively, one would expect that frequent IP switches lead to a significant variation of perceived performance in terms of latency.

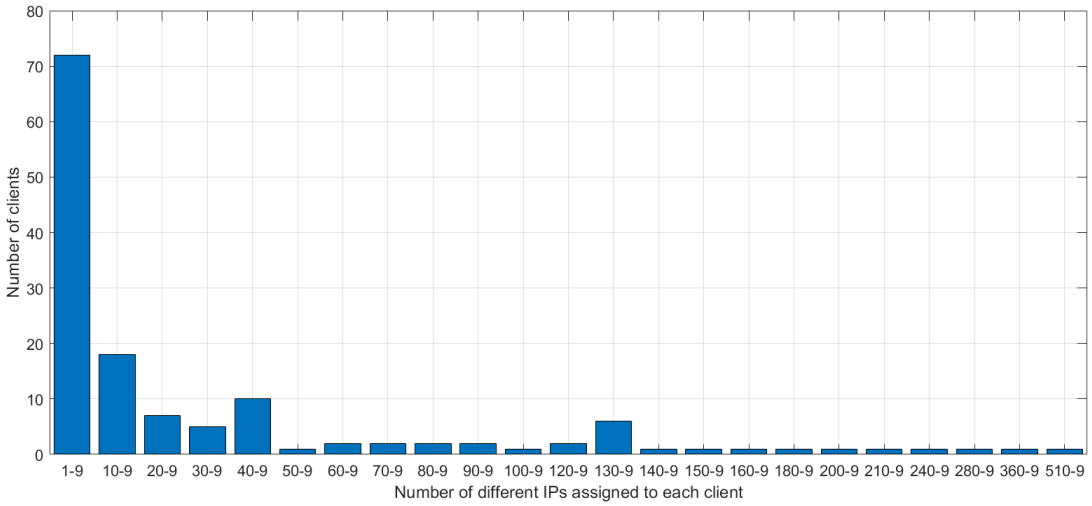
9.1 IP Distribution

We analyse the distribution of different IPs among mobile clients, both on cellular and WiFi connections. More precisely, we look at the number of different assigned IPs per client. Figure 9.1 shows the outcome of this analysis. We observe that significantly more IP addresses are assigned to cellular clients than to WiFi users. Indeed, the most dynamic client on cell uses between 510 and 519 IP addresses, while the most dynamic one on WiFi only between 190 and 199! Nevertheless, for both kinds of clients, we note that more than 50% of them have been assigned less than 20 IP addresses.

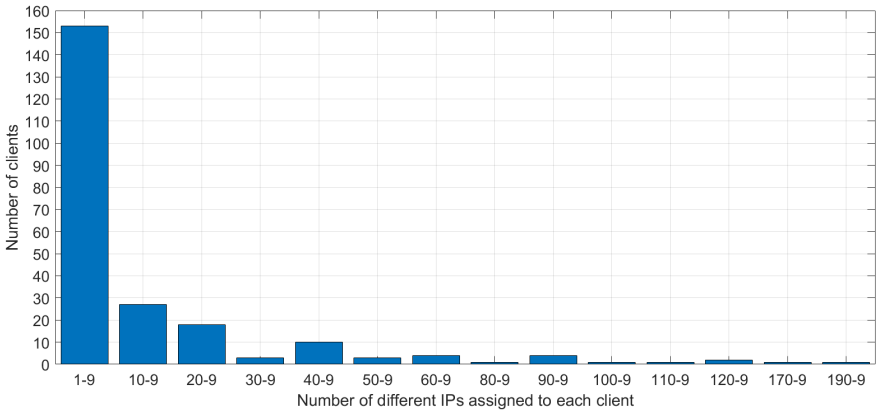
There are several factors which can lead to an IP address change: outages, device reboots and reconnects, and periodic changes imposed by ISP policies [66]. The dynamic behaviour for mobile devices can additionally be explained by the fact that a smart device changes its IP address as soon as it enters a different network. As the main characteristic of cell clients is their mobility, the latter plays an important role when it comes to IP assignment dynamics in cellular networks. Moreover, a device might change its IP in case it switches to another network technology (from 4G to 3G, for example), but this depends on the policy employed by the ISP.

We next take a closer look at the IP assignment dynamics in cellular networks. In particular, we analyse whether the clients remain in the same sub-network/AS/organisation when they start using another IP. Table 9.1 summarises our results. We note that there is no apparent correlation between the number of assigned IPs and the number of sub-networks the clients are hosted in. This is well exemplified by the 18 cellular clients with 10 to 19 different IP addresses: for some of these clients, all of their IPs are in the same prefix, whereas for others the IPs are scattered through multiple subnets. Also note the clients with 180 – 189 and 200 – 209 IPs: the client with less than 190 IPs has visited nearly twice as many prefixes as the one with more than 200 IPs! The same observation holds for the number of hosting ASes and organisations. Moreover, it is interesting to see that the numbers of hosting ASes and organisations are identical: this indicates that the different ASes seen by a client are most often owned by different organisations, which might seem surprising.

Furthermore, we examine whether the dynamic cellular IP distribution is a characteristic of some specific ASes, and we find that it does not seem to be the case. Indeed, numerous ASes host both kinds of cellular clients, i.e. users with a lot of distinct IP addresses and users with less than ten IPs. We do observe some isolated ASes, such as AS 12430 (Vodafone, Spain) and AS 44244 (Irancell, Iran), which exclusively host clients with a high number of different IPs. However, our CELL dataset contains only one client for each of these ASes, which is clearly not enough to draw conclusions.



(a) Number of different IPs per client on cell.



(b) Number of different IPs per client on WiFi.

Figure 9.1: Number of different IPs assigned to each client.

We additionally study the correlation between the number of assigned IP addresses and the number of experienced IP switches on cellular networks. Figure 9.2 confirms that these two metrics present a high positive linear correlation. This indicates that a client using a relatively limited number of IPs does not frequently switch back and forth between them.

A last aspect we investigate is the frequency of IP switches, i.e. how often a cellular client switches from one IP address to another. We perform this analysis on a daily, weekly, and monthly basis and focus on the top 10 clients in terms of number of different IP addresses. Our results are depicted in Figure 9.3. Each curve corresponds to one client. From Figure 9.3a, we can see that 70% of the clients experience more than 100 IP changes in at least one month. Nevertheless, we observe through Figures 9.3b and 9.3c that users change their IP far less frequently during one week/day than during one month: 60% of them switch between different IP addresses less than 40 times per week and at most ten times per day. This leads us to the assumption that the IP changes are performed on a rather regular basis throughout the month.

# diff. IPs per client	# clients	different # subnets per client	# different ASes per client	# different organisations per client
1-9	72	{1, 2, 3, 5}	{1, 2}	{1, 2}
10-19	18	{1, 2, 3, 4, 5, 6, 7, 8, 11}	{1, 2, 3}	{1, 2, 3}
20-29	7	{1, 2, 3, 5, 11, 23}	{1, 2}	{1, 2}
30-39	5	{1, 3, 4, 20}	{1}	{1}
40-49	10	{1, 2, 4, 6, 11, 18, 21}	{1, 2, 3}	{1, 2, 3}
50-59	1	{1}	{1}	{1}
60-69	2	{6, 7}	{1}	{1}
70-79	2	{1, 37}	{1}	{1}
80-89	2	{1, 6}	{1}	{1}
90-99	2	{2, 11}	{1, 2}	{1, 2}
100-109	1	{5}	{1}	{1}
120-129	2	{11, 27}	{1, 2}	{1, 2}
130-139	6	{3, 4, 7, 10, 12, 57}	{1, 2, 3}	{1, 2, 3}
140-149	1	{8}	{2}	{2}
150-159	1	{4}	{1}	{1}
160-169	1	{8}	{1}	{1}
180-189	1	{49}	{1}	{1}
200-209	1	{28}	{1}	{1}
210-219	1	{2}	{1}	{1}
240-249	1	{6}	{1}	{1}
280-289	1	{4}	{1}	{1}
360-369	1	{13}	{1}	{1}
510-519	1	{27}	{2}	{2}

Table 9.1: Prefix/AS/organisation dynamics. The elements of the sets are the different values observed among our cellular clients (i.e. each encountered value is reported only once).

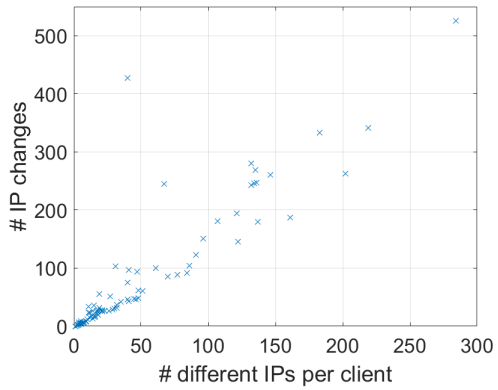
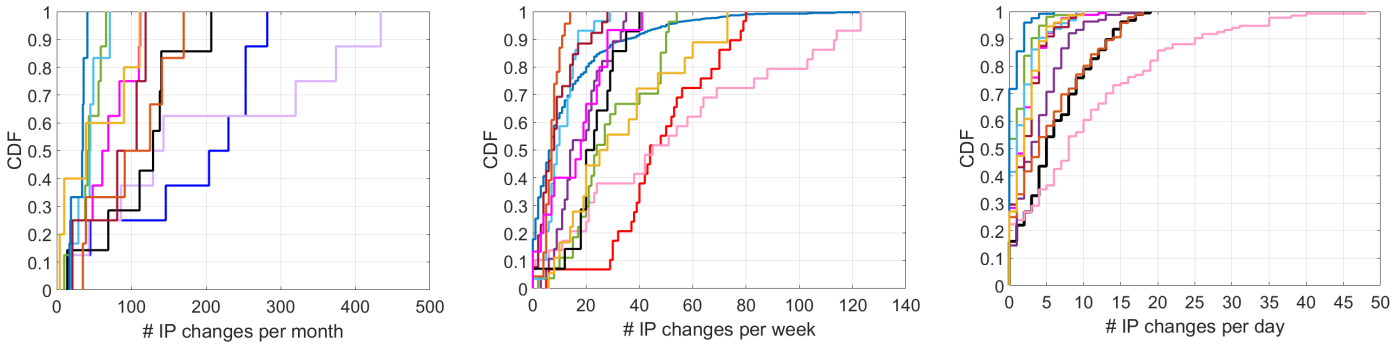


Figure 9.2: Correlation between the number of different IPs assigned to each cellular client and the number of IP changes.



(a) Number of monthly IP switches.

(b) Number of weekly IP switches.

(c) Number of daily IP switches.

Figure 9.3: Number of IP switches on cell.

9.2 Impact on Anycast Performance

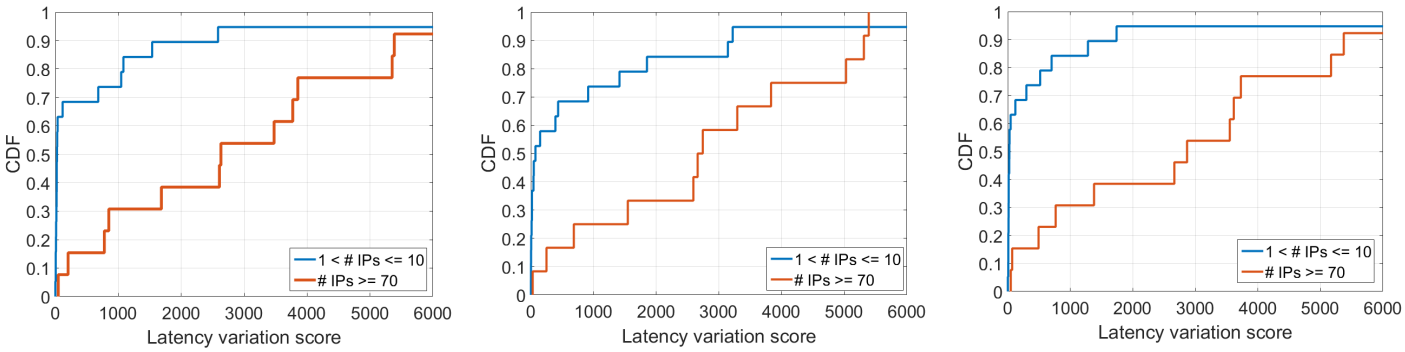
We investigate whether IP dynamics have an impact on the performance of anycast in terms of latency. In particular, we analyse the variation of the anycast latency of the three DNS services in cellular networks. Indeed, variation of latency can have a negative influence on the Quality of Experience (QoE) of the user: she most probably will consider her Internet experience as bad in case she can sometimes access content very fast, but often only slowly.

For this part of our study, we proceed as follows: for each client, we consider their different IPs. For every IP of the given client, we consider all minimum RTTs gathered with the `ping` measurements having this address as source and the considered anycast IP as destination, and compute next the median RTT from all these minimum RTTs. We therefore end up with as many median RTTs as IPs for our client. To assess the latency variation experienced by the user, we finally define the *latency variation score*:

$$\text{latency variation score} = (\max_{\text{medians}} - \min_{\text{medians}}) \times \frac{\#\,\text{diff IPs}}{\#\,\text{IP switches} + 1}$$

where $\max_{\text{medians}}$ refers to the maximum median RTT of the client, $\min_{\text{medians}}$ to the minimum median RTT, $\#\,\text{diff IPs}$ to the number of different IPs the client has been assigned to, and $\#\,\text{IP switches}$ to the number of IP changes experienced by the given user. This score is equal to zero in case the client only uses a single IP (as $\max_{\text{medians}} - \min_{\text{medians}}$ equals zero). The fraction $\frac{\#\,\text{diff IPs}}{\#\,\text{IP switches}+1}$ takes into account the IP assignment dynamics: it is equal to one if, for each IP change, the user switches to an IP address never encountered before, and to less than one if the cellular client switches sometimes between the same IP addresses. This evaluation is performed for each of three DNS services separately. We examine two sets of clients: the more *static* ones, i.e. clients relying only on a relatively small number of different IP addresses, and the more *dynamic* ones, i.e. the clients having used a lot of different IPs. We consider a client as static if she uses at most ten distinct IPs; a dynamic client has been assigned at least 70 IPs. This subdivision of our clients allows us to have approximately the same number of users in both sets. In the context of this study, we evaluate the dynamism of a client based on the number of different IP addresses and not on the number of IP switches. However, as discussed in Section 9.1, both metrics are positively linearly correlated, which implies that they are equally relevant for selecting the cellular clients to analyse.

Figure 9.4 illustrates the obtained latency variation scores for K-Root (Figure 9.4a), F-Root (Figure 9.4b), and Google DNS (Figure 9.4c). We can easily note that the latency variation score is in general higher for dynamic clients than for the static ones, for all of the three DNS services. This underlines the fact that clients connecting to the Internet with numerous IP addresses experience non negligible variations in terms of anycast latency. Nevertheless, there are no significant differences between the three anycast services.



(a) Latency variation scores for K-Root.

(b) Latency variation scores for F-Root.

(c) Latency variation scores for Google DNS.

Figure 9.4: Latency variation scores obtained for the three analysed DNS services on cell.

Chapter 10

Conclusions

In this Master's thesis, we presented the first study of anycast performance for mobile users. To this end, we used data collected from a crowd-sourced platform. In particular, our evaluation focused on three distinct anycast services: K-Root, F-Root, and Google DNS. Moreover, we investigated the IP assignment dynamics of mobile clients, i.e. the frequency at which a different IP is assigned to them, and their potential impact on perceived anycast latency variations.

We found that mobile clients are often mapped to a suboptimal anycast replica in terms of latency with respect to closer unicast servers. Moreover, our results demonstrate that clients are very frequently routed to a replica hosted on a site being geographically far away from them. We observed that long distances towards the assigned anycast server are not simply due to the fact that there is no closer unicast replica, and that this phenomenon is, as the latency issue, not bound to specific regions or particular ASes. Additionally, our investigations highlight three major classes of anycast anomalies, namely distant client packet gateways, poor anycast routing within Tier-1 networks, and improper routing out of cellular networks. Finally, our analysis of IP assignment dynamics confirms our intuition. Indeed, the more dynamic clients, i.e. the ones numerous unique IP addresses have been assigned to, experience a much more pronounced variation in terms of latency, independently of the DNS service. Furthermore, we noted that there is a positive linear correlation between the number of assigned IPs and the number of IP changes per client. This part of the study also reveals the fact that cellular clients are assigned more distinct IP addresses than WiFi users, which can partially be explained by the greater mobility of cell clients.

10.1 Future Work

There are several lines of research arising from this work which are worth being pursued.

IPv4 vs. IPv6 For now, we have only looked at anycast using IPv4. A logical next step would be to launch the same kind of measurement campaign, but relying on IPv6. As there are some striking differences between both versions of the IP protocol, it would be very relevant to compare their behaviour when confronted to the anycast paradigm. Unfortunately, we could not yet launch this analysis on a broad scale. Indeed, only very few of our ALICE clients have IPv6 enabled, and we would not have been able to collect data to draw meaningful conclusions.

AS relationships Looking at inter-AS relationships could greatly aid in explaining some of the encountered phenomena. Indeed, packet routing is highly influenced by ISP policies which are set such that the corresponding ISP has to pay as little as possible. Concretely, this means that a provider prefers that a client's packet travels over a longer AS path consisting in nearly only peer-to-peer links (unpaid exchange of traffic) than over a shorter AS path containing customer-provider links (in this case, the provider has to pay to forward her traffic over the neighbour AS).

Mobile anycast in CDNs In this work, we exclusively focused on mobile anycast in the DNS infrastructure. It would be very interesting to analyse the behaviour of anycast for mobile clients in a Content Delivery Network. Specifically in this direction, we are in the process of establishing a collaboration with EdgeCast Networks, a CDN owned by Verizon Digital Media Services. Unfortunately, the negotiation process was slowed down by legal issues, which prevented us to present first results in this thesis.

In-depth analysis of IP assignment dynamics The study presented in Chapter 9 is a first analysis of the impact of IP assignment dynamics on performance variation. However, our dataset was not very appropriate for this kind of evaluation. Indeed, as explained in Chapter 6, measurements are launched opportunistically, implying that they are not necessarily issued on a regular basis. This makes an in-depth study of IP assignment dynamics difficult, as we would need frequent measurements to get a very good picture of how IPs changes and what the real consequences of these switches are. One possibility would be to consider only a few mobile clients whom we would monitor in a very detailed manner. Instead of using ALICE, we also considered trying out the Mobilyzer platform [67].

Bibliography

[1] J. Abley and K. Lindqvist, "Operation of Anycast Services," Internet Requests for Comments, RFC Editor, RFC 4786, December 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4786.txt

[2] R. de O. Schmidt, J. Heidemann, and J. H. Kuipers, "Anycast Latency: How Many Sites Are Enough?" in *Proceedings of the 2017 Passive and Active Measurement Conference*, March 2017.

[3] J. Hiebert, P. Boothe, R. Bush, and L. Lynch, "Determining the Cause and Frequency of Routing Instability with Anycast," in *Proceedings of the Second Asian International Conference on Technologies for Advanced Heterogeneous Networks*, November 2006.

[4] P. Bret, K. Prashanth, J. Samir, and A. K. Zaid, "TCP over IP Anycast – Pipe dream or Reality?" September 2015. [Online]. Available: https://engineering.linkedin.com/network-performance/tcp-over-ip-anycast-pipe-dream-or-reality

[5] R. Bellis, "Researching F-root Anycast Placement Using RIPE Atlas," October 2015. [Online]. Available: https://labs.ripe.net/Members/ray_bellis/researching-f-root-anycast-placement-using-ripe-atlas

[6] J. H. Kuipers, "Analysing the K-root Anycast Infrastructure," in *25th Twente Student Conference on IT*, July 2016.

[7] L. Wei and J. Heidemann, "Does Anycast Hang up on You?" in *Proceedings of the Network Traffic Measurement and Analysis Conference 2017*, June 2017.

[8] Ericsson, "Ericsson Mobility Report – On the Pulse of the Networked Society," June 2016. [Online]. Available: https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf

[9] Ofcom, "The Communications Market Report," December 2016. [Online]. Available: https://www.ofcom.org.uk/__data/assets/pdf_file/0026/95642/ICMR-Full.pdf

[10] D. McPherson, D. Oran, D. Thaler, and E. Osterweil, "Architectural Considerations of IP Anycast," Internet Requests for Comments, RFC Editor, RFC 7094, January 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7094.txt

[11] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye, "Analyzing the Performance of an Anycast CDN," in *Proceedings of the 2015 Internet Measurement Conference*, October 2015.

[12] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP–4)," Internet Requests for Comments, RFC Editor, RFC 4271, January 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4271.txt

[13] G. Leduc, "Course "Computer Network Architectures and Multimedia"," 2016-2017. [Online]. Available: http://www.montefiore.ulg.ac.be/~leduc/cours/structure-multimedia.html

[14] F. Chen, R. K. Sitaraman, and M. Torres, "End-User Mapping: Next Generation Request Routing for Content Delivery," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, August 2015.

[15] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, "Dissecting Video Server Selection Strategies in the YouTube CDN," in *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, June 2011.

[16] V. K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z. L. Zhang, M. Varvello, and M. Steiner, "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1984–1997, December 2015.

[17] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," 2016.

[18] P. Casas, P. Fiadino, A. Bär, A. D'Alconzo, A. Finamore, and M. Mellia, "YouTube All Around: Characterizing YouTube from Mobile and Fixed-line Network Vantage Points," in *2014 European Conference on Networks and Communications (EuCNC)*, June 2014.

[19] P. Casas, A. D'Alconzo, P. Fiadino, A. Bär, A. Finamore, and T. Zseby, "When YouTube Does not Work – Analysis of QoE-Relevant Degradation in Google CDN Traffic," *IEEE Transactions on Network and Service Management*, vol. 11, no. 4, pp. 441–457, December 2014.

[20] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. L. Zhang, "Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery," in *2012 Proceedings IEEE INFOCOM*, March 2012.

[21] C. Contavalli, W. Van der Gaast, D. Lawrence, and W. Kumari, "Client Subnet in DNS Queries," Internet Requests for Comments, RFC Editor, RFC 7871, May 2016. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7871.txt

[22] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe, "Anycast CDNs Revisited," in *Proceedings of the 17th International Conference on World Wide Web*, April 2008.

[23] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe, "A Practical Architecture for an Anycast CDN," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 4, pp. 17:1–17:29, October 2011.

[24] A. Flavel, P. Mani, D. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev, "FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, May 2015.

[25] X. Fan, J. Heidemann, and R. Govindan, "Evaluating Anycast in the Domain Name System," in *2013 Proceedings IEEE INFOCOM*, April 2013.

[26] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky, "Evaluating The Effects Of Anycast On DNS Root Nameservers," in *RIPE 53*, October 2006.

[27] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy, "Two Days in the Life of the DNS Anycast Root Servers," in *Proceedings of the 8th International Conference on Passive and Active Network Measurement*, April 2007.

[28] P. Barber, M. Larson, and M. Kosters, "Traffic Source Analysis of the J Root Anycast Instances," in *NANOG 39*, February 2007.

[29] H. Ballani, P. Francis, and S. Ratnasamy, "A Measurement-based Deployment Proposal for IP Anycast," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, October 2006.

[30] G. C. M. Moura, R. de O. Schmidt, J. Heidemann, W. B. de Vries, M. Muller, L. Wei, and C. Hesselman, "Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event," in *Proceedings of the 2016 Internet Measurement Conference*, November 2016.

[31] S. Sarat, V. Pappas, and A. Terzis, "On the Use of Anycast in DNS," in *Proceedings of 15th International Conference on Computer Communications and Networks*, October 2006.

[32] K.-L. Neoh and H.-T. Ewe, "An Active Anycast Approach to Improve Roaming Mobile IPv6 Client Performance in Multi-server Environment," in *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, September 2008.

[33] P. I. Basarkod and S. S. Manvi, "Mobility and QoS aware anycast routing in Mobile ad hoc Networks," *Computers & Electrical Engineering*, vol. 48, pp. 86–99, November 2015.

[34] S.-K. Chen and P.-C. Wang, "Design and Implementation of an Anycast Services Discovery in Mobile Ad Hoc Networks," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 6, no. 1, pp. 2:1–2:9, February 2011.

[35] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," Internet Requests for Comments, RFC Editor, RFC 3775, June 2004. [Online]. Available: https://www.ietf.org/rfc/rfc3775.txt

[36] D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Augé, and T. Friedman, "A Fistful of Pings: Accurate and Lightweight Anycast Enumeration and Geolocation," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015.

[37] ——, "Latency-Based Anycast Geolocalization: Algorithms, Software and Data Sets," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1889–1903, May 2016.

[38] D. Madory, C. Cook, and K. Miao, "Who are the Anycasters?" in *NANOG 59*, October 2013.

[39] D. Cicalese, J. Augé, D. Joumblatt, T. Friedman, and D. Rossi, "Characterizing IPv4 Anycast Adoption and Deployment," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, December 2015.

[40] Belgian Institute for Postal services and Telecommunications, "Coverage maps: mobile networks," January 2017. [Online]. Available: http://www.bipt.be/en/consumers/telephone/quality-of-service/coverage-maps-mobile-networks

[41] P. Mockapetris, "Domain Names – Concepts and Facilities," Internet Requests for Comments, RFC Editor, RFC 1034, November 1987. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1034.txt

[42] Information Sciences Institute, "B-Root begins Anycast in May," April 2017. [Online]. Available: http://www.root-servers.org/news/b-root-begins-anycast-in-may.txt

[43] D. Karrenberg, "DNS Root Name Server FAQ," January 2005. [Online]. Available: http://www.isoc.org/briefings/020_v1/briefing20.pdf

[44] A. Pacific Network Information Centre, "Root servers – FAQs." [Online]. Available: https://www.apnic.net/get-ip/faqs/rootservers/

[45] Microsoft, "Disable Recursion on the DNS Server." [Online]. Available: https://technet.microsoft.com/en-us/library/cc771738(v=ws.11).aspx

[46] G. Kessler and S. Shepard, "A Primer On Internet and TCP/IP Tools and Utilities," Internet Requests for Comments, RFC Editor, RFC 2151, June 1997. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2151.txt

[47] J. Postel, "Internet Control Message Protocol," Internet Requests for Comments, RFC Editor, RFC 792, September 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc792.txt

[48] J. Messer, *Secrets of Network Cartography: A Comprehensive Guide to Nmap.* NetworkUptime.com, 2008.

[49] J. Postel, "User Datagram Protocol," Internet Requests for Comments, RFC Editor, RFC 768, August 1980. [Online]. Available: http://www.rfc-editor.org/rfc/rfc768.txt

[50] ——, "Transmission Control Protocol," Internet Requests for Comments, RFC Editor, RFC 793, September 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[51] G. Malkin, "Traceroute Using an IP Option," Internet Requests for Comments, RFC Editor, RFC 1393, January 1993. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1393.txt

[52] M. Luckie, Y. Hyun, and B. Huffaker, "Traceroute Probe Method and Forward IP Path Inference," in *Proceedings of the 2008 Internet Measurement Conference*, October 2008.

[53] Northwestern University – Aqualab, "ALICE – Mobile Experiment Engine." [Online]. Available: http://aqualab.cs.northwestern.edu/projects/261-alice

[54] J. P. Rula, "Adopting a Gateway Centric View for Cellular Network Content Delivery," Ph.D. dissertation, Northwestern University, 2016.

[55] Northwestern Undergraduate User Experience and Mobile Development Team, "NU Signals v2." [Online]. Available: https://nux.northwestern.edu/projects/nu-signals-v2

[56] Northwestern University – Aqualab, "Application Time (AppT)." [Online]. Available: http://www.aqualab.cs.northwestern.edu/projects/283-appt

[57] ——, "Namehelp Mobile." [Online]. Available: http://aqualab.cs.northwestern.edu/projects/237-namehelp-mobile

[58] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian, "Where's That Phone?: Geolocating IP Addresses on 3G Networks," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, November 2009.

[59] T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations," *Survey Review*, vol. 23, no. 176, pp. 88–93, 1975.

[60] D. J. Griffiths, *Introduction to Electrodynamics*. PHIL, 2012.

[61] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 53–56, April 2011.

[62] IP2Location, "IP2Location LITE IP-COUNTRY-REGION-CITY-LATITUDE-LONGITUDE Database." [Online]. Available: http://lite.ip2location.com/database/ip-country-region-city-latitude-longitude

[63] Cisco, "Understanding Delay in Packet Voice Networks," February 2006. [Online]. Available: http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/5125-delay-details.html

[64] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. M. Mao, "Cellular Data Network Infrastructure Characterization and Implication on Mobile Content Placement," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, June 2011.

[65] L. Zhihao and N. Spring, "Tier-1's break Anycast DNS," in *Workshop on Active Internet Measurements (AIMS) 2017*, March 2017.

[66] R. Padmanabhan, A. Dhamdhere, E. Aben, K. Claffy, and N. Spring, "Reasons Dynamic Addresses Change," in *Proceedings of the 2016 Internet Measurement Conference*, November 2016.

[67] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao, "Mobilyzer: An Open Platform for Controllable Mobile Network Measurements," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, May 2015.