ABSTRACT

Title of dissertation: DIAGNOSING AND IMPROVING THE PERFORMANCE OF INTERNET ANYCAST

Zhihao Li
Doctor of Philosophy, 2019

Dissertation directed by: Professor Neil Spring
Department of Computer Science

IP anycast is widely used in Internet infrastructure, including many of the root and top-level DNS servers, major open DNS resolvers, and content delivery networks (CDNs). Increasing popularity of anycast in DNS resolvers involves it in most activities of Internet users. As a result, the performance of anycast deployments is critical to all the Internet users.

What makes IP anycast such an attractive option for these globally replicated services are the desired properties that anycast would appear to achieve: reduced overall access latency for clients, improved scalability by distributing traffic across servers, and enhanced resilience to DDoS attacks. These desired properties, however, are not guaranteed. In anycast, a packet is directed to certain anycast site through inter-domain routing, which can fail to pick a route with better performance in terms of latency or load balance. Prior work has studied anycast deployments and painted a mixed picture of anycast performance: many clients of anycast are not served by their nearby anycast servers and experience large latency overheads;

anycast sometimes does not balance load across sites effectively; the catchment of an anycast site is mostly stable, but it is very sensitive to routing changes.

Although it was observed over a decade ago that anycast deployments can be inefficient, there exist surprisingly few explanations on the causes or solutions. In addition, most prior work evaluated only one or several deployments with measurement snapshots. I extended previous studies by large-scale and longitudinal measurements towards distinct anycast deployments, which can provide more complete insights on identifying performance bottlenecks and providing potential improvements. More importantly, I develop novel measurement techniques to identify the major causes for inefficiency in anycast, and propose a fix to it. In this dissertation, I defend the following thesis: *Performance-unawareness of BGP routing leads to larger path inflation in anycast than in unicast; and with current topology and protocol support, a policy that selects routes based on geographic information could significantly reduce anycast inflation.*

In the first part of the dissertation, I use longitudinal measurements collected from a large Internet measurement platform towards distinct anycast deployments to quantitatively demonstrate the inefficiency in performance of anycast. I measured most root DNS servers, popular open DNS resolvers, and one of the major CDNs. With the passive and active measurements across multiple years, I illustrate that anycast performs poorly for most deployments that I measured: anycast is neither effective at directing queries to nearby sites, nor does it distribute traffic in a balanced manner. Furthermore, this longitudinal study over distinct anycast deployments shows that the performance has little correlation with number of sites.

In the second part of the dissertation, I focus on identifying the root causes for the performance deficits in anycast. I develop novel measurement techniques to compare AS-level routes from client to multiple anycast sites. These techniques allow me to reaffirm that the major cause of the inefficiency in anycast is the performance-unawareness of inter-domain routing. With measurements from two anycast deployments, I illustrate how much latency inflation among clients can be attributed to the policy-based performance-unaware decisions made by BGP routing. In addition, I design BGP control plane experiments to directly reveal relative preference among routes, and how much such preference affects anycast performance. The newly discovered relative preferences shed light on improving state-of-art models of inter-domain routing for researchers.

In the last part of the dissertation, I describe an incrementally deployable fix to the inefficiency of IP anycast. Prior work has proposed a particular deployment scheme for anycast to improve its performance: anycast servers should be deployed such that they all share the same upstream provider. However, this solution would require re-negotiating services that are not working under such a deployment. Moreover, to put the entire anycast service behind a single upstream provider introduces a single point of failure. In the last chapter, I show that a static hint with embedded geographic information in BGP announcements fixes most of the inefficiency in anycast. I evaluate the improvements from such static hints in BGP route selection mechanisms through simulation with real network traces. The simulation results show that the fix is promising: in the anycast deployments I evaluated, the fix reduces latency inflation for almost all clients, and reduces latency by 50ms for

23% to 33% of the clients. I further conduct control plane experiments to evaluate the effectiveness of the static hints in BGP announcements with real-world anycast deployments.

This dissertation provides broad and longitudinal performance evaluation of distinct anycast deployments for different services, and identifies an at-fault weakness of BGP routing which is particularly amplified in anycast, i.e., route selection is based on policies and is unaware of performance. While applying the model of BGP routing to diagnose anycast, anycast itself serves as a magnifying glass to reveal new insights on the route selection process of the BGP in general. This work can help refine the model of route selection process that can be applied to various BGP-related studies. Finally, this dissertation provides suggestions to the community on improving anycast performance, which thus improves performance and reliability for many critical Internet infrastructure and ultimately benefits global Internet users.

DIAGNOSING AND IMPROVING THE PERFORMANCE OF INTERNET ANYCAST

by

Zhihao Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Neil Spring, Chair/Advisor
Professor Bobby Bhattacharjee
Professor Dave Levin
Professor Richard La
Professor Tudor Dumitraș
Professor Mark Shayman

Acknowledgments

I owe my gratitude to all the people who advised, supported, encouraged and inspired me during my Ph.D. life. I will keep in mind all achievements and lessons from my graduate study.

I would first like to thank my advisors, Prof. Neil Spring, for giving me the great opportunity to explore research topics in understanding the Internet. His broad knowledge in the domain and insightful ideas have helped me shape research problems and develop solutions in so many work, and calibrated me on the right direction in research. His passion for research and attention to details are qualities that always motivate me and I have been learning from. His encouragement and patience leads me through many difficult times in research. Despite the enormous amount of work on the new building, Neil has been always accommodative to discuss research with me. Without advice from him, this dissertation would not be possible.

I would also like to thank Prof. Bobby Bhattacharjee and Prof. Dave Levin for advising me. Bobby's inspiring questions and feedbacks have helped this dissertation significantly. He has taught me how to write papers, gave me training on giving presentations and addressing questions in talks. I also have worked closely with Dave and learned from him throughout my graduate life. I benefit a lot from his organized approach to tackle research problems, his amazing skills in presentation and talks, and his great creativity and sense of humor. His energy and passion on research motivates me to expand my research. Bobby and Dave helped me grow as a researcher. I am truly grateful to both of them.

I also would like express my appreciation to Prof. Richard La, Prof. Tudor Dumitras and Prof. Mark Shayman for serving on my thesis committee and for the influential discussions and feedbacks.

I am very grateful to my colleagues and lab-mates. Ramakrishna Padmandabhan has been a great colleague and friend, he helped me a lot with my research and presentations. I will always remember the days and nights we worked on our first paper. I am also grateful to Matthew Lentz for his friendship and support. Matt provided me much help in the GNU Radio project we worked on together, and has been always helpful for my presentations. I will keep many his "life advice" in mind. I feel fortunate to work with Stephen Herwig in various projects. I would like to thank him for his warm support and friendship during some difficult times in my graduate life. Youndo Lee is a friend and mentor to me, I thank him for teaching me research in early days. Thank you, James, Katura, and Richie, for your help and encouragement. Thank you all of my lab-mates for making the lab a great place.

I would like to thank my friends, who have made my life in graduate school a pleasant experience: Wei Bai, Yaming Wang, Doowon Kim, Min Ye, Heng Zhang, Heng Qiao, Ang Gao, Wentao Luan and many many more. It has been a pleasure to experience graduate life with you, and I am glad that we shared some of the more memorable moments.

Last but not least, I owe my deepest gratitude to my parents for their generous support and continuous love. Thank you my mom and dad for all the encouragement and always having faith in me. Thanks to my love, Weiwei, who has always been there for me.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Anycast is one of the main addressing methods in the Internet Protocol, and is increasingly used in major network infrastructure. It provides an *one-to-any* association where the packets are routed to any one of a group of receivers that are all identified by the same destination IP address. Anycast is especially popular in Domain Name System (DNS) and content delivery networks (CDNs). DNS is the service that translates human-readable domain names to IP addresses and vice versa. It is involved in various activities that Internet users do today: browsing web pages, communicating with each other, conducting online transactions and accessing emergency services [1, 2]. As of today, all 13 DNS root servers and many popular open DNS resolvers from providers like Google [3], openDNS [4] and Cloudflare [5], are hosted via IP anycast. Some content delivery networks use anycast in an attempt to lower latencies and distribute load better. For example, popular sites like Stack Overflow and Yelp are hosted on Cloudflare's anycast-based CDNs [5]. As a result, the performance of anycast deployments is critical to almost all Internet users.

Large-scale and longitudinal measurements towards anycast deployments can help network operators and service administrators to evaluate and understand how

well their anycast is working, identify performance bottlenecks and potential improvements. For example, network attacks, especially distributed Denial-of-Service (DDoS) attacks can dramatically degrade the performance of the Internet infrastructure, including anycast-based infrastructure [6]. Comprehensive measurements can inform us how did the anycast deployments react to DDoS attacks with different volumes, and help to identify the most affected Internet users. Comparing measurements towards different anycast deployments under similar DDoS attacks can provide insights on potential improvements. For example, operators should deploy more hosts at the anycast sites under large pressures during DDoS. And measurements after changes in deployments can help reveal how effective the changes are in improving anycast performance [7].

The insights obtained from the measurements towards different anycast deployments can benefits multiple parties around the Internet ecosystem, including service operators, Internet Service Providers (ISPs), and users themselves. Service operators, especially DNS service operators, constantly monitor the quality of their services [8–10]. They would like to identify any problems around their service and fast react to them. Also when planning new anycast sites in their deployments, service operators use passive and active measurements to understand how to effectively setup new servers [7,11]. ISPs, especially large transit providers, are more critical to the performance of anycast deployments than they might be aware of. These measurements can help ISPs identify underlying problems in their local routing policies that adversely impacts anycast performance. Ultimately, the insights gained from measuring anycast performance will benefit most Internet users, as their Internet

activities would probably involve interactions with anycasted services.

Prior work measured anycast-based services, but most of them focused on only one or several deployments. We previously lack broad and longitudinal analysis of different anycast deployments. Several prior studies have shown that anycast can be inefficient, but much to our surprise, there exists few explanations of the causes of inefficiency or solutions to it.

1.1 Background: what to expect from IP anycast

In anycast, packets are sent to any one instance of a set of replica servers that are assigned with the same address. The service provided by each replica server is generally equivalent regardless of the site the server is located [12]. For IP anycast in particular, replica servers at two or more geographic anycast sites announce the same IP address through the Border Gateway Protocol (BGP), the de-facto inter-domain routing protocol. Packets are directed to one of the replica servers through BGP routes, and senders have no control over which server receives them.

A service may use anycast for a variety of reasons. Several desired properties of anycast deployments are:

- Performance: Anycast can help to reduce overall access latency for clients, by providing them geographic proximity to reduce network distance between clients and the server.

- Scalability: Anycast infrastructure distributes (coarsely) traffic load across a set of replicas, which allows infrastructure to scale to handle increased traffic

and to accommodate traffic peaks.

- Reliability: Anycast can mitigate distributed denial-of-service (DDoS) attacks by constraining attacks to the catchment of anycast sites. The catchment of an anycast site is the topological region of the Internet from which packets towards the anycast address are directed to the site. Anycast catchment also provides information to help identify the sources of attacks where traffic originated with spoofed addresses.

These properties that IP anycast would appear to achieve make it an attractive option for globally replicated service deployments. Anycast is widely used in critical Internet infrastructures, including many of root and top-level DNS servers, major open DNS resolvers, and some content delivery networks (CDNs).

The desired properties of anycast, however, are not guaranteed. The anycast server chosen to provide service to certain clients is determined by BGP routing, which lacks mechanisms to pick routes with better performance in terms of latency or geographic proximity, load balance, and catchment stability. Nonetheless, network operators expect anycast generally achieves the properties, as evidenced by the increasing deployments of root DNS servers and open resolvers [10, 13–15].

In this dissertation, I use the following anycast-related terminologies: *Anycast address* is the IP address announced from different physical locations across the Internet, called *sites.* Each site may have one or more servers, called *replicas.* For a specific anycast deployment, a given site is either *local* or *global.* Replicas at local sites are often announced with *no-export* BGP community to prevent the hosting

AS from announcing them to peers. *Local* replicas are available only within the hosting AS or its customers. *Global* replicas have no such constraints and can be accessed across the Internet.

1.2 Prior approaches and remaining challenges

Prior studies have measured various anycast services, but few provided insights on the root causes of anycast inefficiencies.

In Chapter 2, I place the problem of evaluating and improving IP anycast in the context of related work, and describe the prior results. These work has measured many different anycast services (e.g., DNS roots and CDNs), and evaluated the performance of anycast mainly from three aspects: latency or geographic proximity [7, 9, 16–21], load distribution [6, 11, 18, 20, 22], and catchment stability [16, 18–20, 22, 23]

However, prior results painted a mixed picture of anycast performance: the deployment of anycast reduces overall latency for clients, but many clients were not served by their nearby anycast servers and experienced large latency overheads; while catchment is generally stable in anycast, it is also quite sensitive to routing changes; anycast sometimes does not balance load across instances effectively. Some prior work has suggested the significant impact of BGP routing on anycast performance degrading, but unfortunately, little work has been done to fix the problem.

Ballani et al. [18, 24] claimed that the lack of metrics in route selection mechanisms to identify the better route is the cause for anycast inefficiency, and proposed

one way to fix the inefficiency: anycast instances should be deployed such that they all share the same upstream provider. This solution suggests that cooperation with a large global ISP is a prerequisite to have efficient anycast-based services. For services that are not working under such deployment, it requires negotiating with their providers to implement the fix. Also, this deployment scheme introduces a single point of failure (the single upstream provider) in anycast, and makes anycast more vulnerable to interruptions in the provider.

1.3 Thesis

The goal of this dissertation is to identify the major causes for inefficiency in anycast, and to propose a potential fix. To achieve the goal, I develop active probing methods to obtain large-scale and longitudinal measurements towards various anycast deployments. In this dissertation, I defend the following thesis: *Performance-unawareness of BGP routing leads to larger path inflation in anycast than in unicast; and with current topology and protocol support, a policy that selects routes based on geographic information could significantly reduce anycast inflation.*

- Policy-based performance-unaware Inter-domain routing: Inter-domain routing protocol selects routes to forward packets based mainly on routing policies, and has no mechanisms to identify the routes with better performance.

- Path inflation: The network path taken between two end hosts is longer than necessary.

1.4 Contributions

To demonstrate this thesis, I conduct in-depth analysis of distinct IP anycast deployments (Chapter 3), including different DNS servers and CDNs. I then investigate the prevalent inefficiencies among anycast deployments, and identify the root causes for performance problems (Chapter 4): routes lack of useful information to identify routes to nearby, low-latency anycast sites. I find that ASes usually receive routes that are "equivalent" based on the state-of-art model for inter-domain routing, and the ASes cannot identify routes to closer sites. However, I develop experiments to reveal that relative preference exists among the "equivalent" routes (Chapter 5). Finally, I propose how to improve anycast efficiency and evaluate the benefits with simulations (Chapter 6). My contributions are organized as the following:

Chapter 3: Evaluating anycast performance

Using passive and active measurements of distinct, DNS anycast deployments, I quantify the inefficiency of IP anycast, in terms of both latency and load balance. While prior results showed that IP anycast might not be optimal, I find that over 20% of clients traveled an extra 2000km (over the distances to their closest anycast instances) for most of the measured anycast deployments. Contradicting suggestions in prior work, I use measurements collected from real anycast deployments to show that adding anycast instances often increases overall latency for clients. Further, I describe algorithms to characterize influential routing changes that cause performance shifts in anycast, and a heuristic on identifying the causes of such routing changes.

Chapter 4: Diagnosing anycast performance problems

I develop a novel measurement technique to compare the AS-level paths from vantage points to multiple anycast instances. My results verify that the major cause of anycast inefficiency is the *performance-unawareness* property of inter-domain routing: route selection mechanisms lack of useful information to identify the better route, thus often choose route towards a distant, high-latency anycast instance. For the two anycast deployments that experienced large latency inflation, over 40% and 65% of the clients, respectively, were directed to distant instances due to poor routing selections among routes with equivalent preference based on current model of inter-domain routing.

Chapter 5: Inferring provider preferences via route manipulations

I design control plane experiments and use large-scale data plane measurements to directly identify relative preference among routes. My experiments expose the ISPs behavior on selecting Tier-1 providers when multiple of them provide valid routes. Research on interdomain routing has been relying on Gao-Rexford model [25, 26] When the model cannot determine the favorable route, it is commonly assumed that a route is selected the equally good ones by the "Shortest AS path" policy. My experiments illustrate that ISPs usually have "local preference" towards providers before taking shortest paths. More surprisingly, I show that Tier-2 ISPs have common and consistent preferences among Tier-1 ISPs, even when they have the same provider-customer relations. Another important implication from the result is that "Shortest AS path" policy might be applied less often than people expected. My result shows that only about half of the Tier-2 ISPs choose shorter

paths from Tier-1 providers. Fortunately, such preference does not cause additional inefficiencies in anycast: it causes about 16.7% of the queries to be sent to farther anycast sites, while 19.2% of the queries to closer sites.

Chapter 6: Improving anycast performance

Instead of deploying all anycast instances such that they all share the same upstream provider [18, 24], I propose to add geographic hints in BGP announcements. Through the proposed static hints, BGP routers can identify the route towards (geographically) nearby anycast instance, thus reduce latency inflation. I evaluate the benefits of such static hints by simulating the route selection process with real traces. The results show that this fix reduces latency inflation for almost all clients, and reduces latency by 50 ms for 23% to 33% of the clients in the two anycast deployments evaluated. I further evaluate the propagation of the static hints in the Internet using a real-world anycast deployment [27].

Chapter 7: Conclusion and future work

I conclude this dissertation with a summary of contributions. First, this dissertation provides a comprehensive and longitudinal performance measurements of distinct anycast deployments. Second, this measurement-based study identifies that the performance-unaware route selection policies in BGP are the major causes for inefficiencies in anycast. Although this weakness of BGP routing is known to researchers, it is amplified in anycast scenarios. Furthermore, measuring anycast reveals new insights on route selection process of the BGP. This finding helps the research community refine the model of BGP routing that can be applied to various BGP-related studies. In addition, this dissertation provides suggestions to the

Internet community on improving anycast performance. Ultimately, these contributions help improve performance and reliability of critical Internet infrastructure and thus benefit global Internet users. In the end, I discuss directions for future work in improving anycast deployments.

Chapter 2: Related Work

2.1 Performance measurements of IP anycast

Much prior work has evaluated the performance of anycast from three aspects: latency, catchment stability and load distribution. In those measurement studies, CDNs and DNS root servers are the popular targets because of their importance as fundamental Internet infrastructure, and the fact that anycast is widely used to provide such services.

2.1.1 Path inflation in IP anycast

Several studies have used RTTs as the metric to evaluate IP anycast. They compared the RTTs between clients to the anycast address with the lowest RTT among the RTTs from clients to all anycast sites [7, 9, 16–19].

As early as in 2006, Sarat et al. [16] conducted such measurements to F- and K-root using Planetlab hosts [28]. They showed that the deployment of anycast reduces average query latency, and that majority (over 50%) of the queries were served by anycast servers with low ($<$ 20ms) latency overheads compared to the lowest-latency server. Anycast deployment of the K-root DNS server is studied by

Colitti et al. [9] in the same year with RIPE's traffic measurements service [29], and concluded with similar results. More recently in 2013, Liang et al. [17] applied King latency inference technique [30] to measure latencies between about 20K open recursive resolvers and root DNS servers. Their results, however, showed that about 40% of the resolvers were routed to anycast sites more than 50ms farther away from the lowest latency anycast sites. Calder et al. [19] measured an anycast-based CDN with embedded JavaScript in Bing search responses, and they reported that anycast usually performed well, although it directed 20% of clients to suboptimal sites.

Other studies have used geographic distance as a metric to evaluate how well anycast performs. In 2007, Liu et al. [20] evaluated C-, F- and K-root with two days' DNS traffic, and reported that queries traveled an extra 5000 km longer than to geographically closest anycast site from about 60%, 40% and 40% of the clients for C-, F-, and K-root DNS servers, respectively. Recently in 2015, Kuipers [21] conducted a 10 minute measurement of K-root's anycast performance, and showed that over 45% of clients are not directed to their geographically closest anycast site.

Previous results painted a mixed picture of anycast performance among all the different anycast deployments measured from different sets of vantage points: overall, anycast reduces access latency for clients; however, usually about 30% of clients experienced large latency overheads for using sub-optimal anycast servers. In this dissertation, I perform an in-depth and longitudinal analysis of over 10 distinct anycast deployments, and quantify the inefficiencies of anycast performance.

In much of the prior work [9, 16, 31], people suggested that more anycast instances improve the overall latency for clients. In a most recent study in 2016,

however, Schmidt et al. [7] stated that having "as few as twelve sites" is enough to provide reasonable performance as having many sites. Our results qualitatively support this statement in the sense that adding more anycast instances does not improve, but in fact *harms* the performance in many anycast deployments. We further show that it is a common problem for anycast deployments that they are unable to fully utilize the performance that could be realized, and more importantly identify the root causes of these inefficiencies.

2.1.2 Route stability for IP anycast

Some Internet services (e.g., video streaming) require stateful connections between clients and servers for a long time. Since the routing system determines the anycast instance selection on behalf of clients, for such services to be provided reliably over anycast, route stability is expected.

Internet routing stability is an important property studied extensively in previous work [32–38], including pioneering work by Paxson [39], in which they analyzed routing failures, loops, symmetry and stability using large-scale end-to-end measurements Much prior work evaluated the impact of routing instability on network latency. In 2006, Wang et al. [40] and Pucha et al. [41] presented first work that quantified the effect of routing change on network latency after route convergence. By analyzing a diverse set of end-to-end paths, they concluded that route changes happen frequently between hosts, and most path changes caused small latency change. Schwartz et al. [42–44] conducted large-scale longitudinal measurements on end-to-

end latency change due to routing dynamics, and found similar results. In addition, their longitudinal analysis showed that the (in-)stability of routes remained similar in the years they conducted their experiments. Many methods on identifying and diagnosing route changes were proposed [45–52]. My work differs from the prior work with the focus on the impact of route instability on IP anycast. Our initial results suggest that routing changes caused larger latency variation in anycast: route changes in anycast could lead clients to distant anycast instances.

More specific to anycast, other prior work studied the catchment stability to anycast sites: how often clients change their directed sites. These studies [11, 16, 18–20, 22] used a special type of DNS queries—*hostname.bind.* TXT record—to identify the anycast instance used by each client, and detected anycast instance swaps. A similar conclusion was drawn from the prior work: IP anycast typically offers stable catchment to most clients; only a small fraction ($< 5\%$) of clients experience frequent swaps, usually due to per-packet load balancing. In a recent study in 2018, Wei et al. [23] confirmed the common understanding that catchment is generally stable in anycast, and identified a small set of clients that are affected by per-packet load balancing. Hiebert et al. [53, 54] proposed to use catchment stability in DNS anycast as an indicator for detecting routing instability. Their evaluation showed their method to be promising: about 61% of the significant routing events were detected by instance swaps in anycast. We did not try to re-examine the catchment stability of anycast, nor to use anycast to detect route instability. With longitudinal analysis on different anycast deployments, we characterize the impact of routing changes on anycast performance, and identify the root causes for the

routing changes in Chapter 3.

2.1.3 Load distribution among anycast sites

Anycast is expected to distribute (coarsely) traffic load across instances, and thus to handle traffic peaks and to mitigate DDoS attacks. In 2006, Ballani et al. [18] showed that IP anycast does not balance client load across anycast instances, and proposed a mechanism through AS-Path prepending and traffic engineering with the anycast providers to coarsely control the traffic distribution. Other work [20,22] examined server logs of DNS roots, and observed very large variability in traffic loads on anycast servers. de Vries et al. [11] provided a new approach to measure anycast catchment and estimate traffic load across anycast instances. Recently in 2016, Moura et al. [6] studied anycast under the pressure of a particular DDoS attack against DNS roots in November and December 2015. They showed that although anycast is overall resilient to DDoS attacks, some anycast instances can become overloaded and result in collateral damage to other instances. Our results largely reinforce these prior results by showing that anycast does not balance load effectively. In Chapter 5, I conduct an experiment that examine if certain Tier-1 ISPs are preferred by customer ISPs, which might causes queries through the Tier-1 ISPs are "funneled" to few anycast sites, thus cause out-of-balance load distribution.

2.2 Explaining and fixing IP anycast performance

Many of the above measurement studies suggest that BGP routing has large impact on whether clients receive low-latency, stable services, but have provided little explanation or fix. A case study from Bellis [55] used active measurements to identify and fix a specific latency issue in F-root caused by route leaks. Ballani et al. [18] is the closest work to ours in identifying the root causes of large latency inflation (or what they refer to as the "stretch-factor") for anycast clients; they claimed that current route selection mechanisms lack of metrics to identify the route with low latency, thus have a high chance of making poor choice of anycast server. Our measurement study on multiple anycast deployments verifies their claim. Furthermore, our results quantify the effects of poor routing decisions on latency inflation, and attribute latency inflation to incurred by each of the common routing policies (e.g., prefer customer-over-peer-over-provider, prefer shortest AS Path).

Ballani et al. [18, 24] hypothesized that deploying the anycast instances such that they all share the same upstream provider is one approach to account for route selection problem, and thus remedy latency inflation in IP anycast. We find that C-root has such deployment, and confirm this hypothesis by showing that C-root clients do not suffer from the poor route selection issues. However, most anycast services are not deployed in this fashion, and changing deployments requires significant amount of work in negotiating with providers. More importantly, deploying anycast instances under a single upstream provider introduces a single point of failure, makes anycast less resilient to DDoS attacks. In Chapter **??**, we propose a easily deploy-

able fix: adding static hints in BGP announcements, that enables BGP's ability to identify the route towards nearby anycast instance. Our evaluation shows that such static hints improve latency performance for most clients.

There are other proposals [24, 56–58] on improving anycast to achieve better load balancing among instances. In this dissertation, we do not directly address the load balance problem in anycast, rather we conduct experiments in attempts to identify the cause of unbalanced load. Furthermore, our proposed static hints in BGP improves geo-proximity for the clients, which ensures traffic is distributed to nearby instances.

2.3 Interdomain routing models and route manipulations

Anycast relies on interdomain routing to direct packets to nearby replicas. Much prior work studied on how to model interdomain policies. Huston [59, 60] presented seminal work on classifying relations between ISPs and introduced the economic considerations in interdomain routing policy. Griffin et al. [61, 62] studied the problem of identifying stable paths by modeling BGP policies and connectivities. The current model of interdomain routing policies was developed by Gao and Rexford [25, 26] based on prior work. This model classifies relations between ISPs into customer-provider, where the customer pays the provider, and peer-peer, where the peers exchange traffic without cost. Also this model provides insights on local preferences based on the costs: An ISP prefers routes through a neighboring customer, then routes through a neighboring peer, and then routes through a providers; i.e., ISPs

prefer cheaper routes. This model has been used to simulate interdomain routing paths selections in myriad studies on analyzing network reliability [63], BGP convergence [64], control of network traffic [65–68], and BGP security [69–71]. Many tools like BSIM [72], BGPSIM [73], QUICKSAND [74] were developed assuming the Gao-Rexford model to predict BGP routing paths. In these simulation tools, the shortest paths among the ones satisfying the model are assumed preferable, and other tie-breakers might be applied to determine the unique path. In this dissertation, I assess how often the rule of preferring shortest paths is actually used by ISPs when selecting next-hop providers.

While used in many studies, it is known that the Gao-Rexford misses some aspects of the interdomain routing. Researchers tried to improve the model by identifying hybrid and partial transit relationships between ISPs [75], providing finer granularity for preference ranking of neighboring ISPs [76], addressing the effect of intra-domain routing policies on interdomain routes [77], etc. Gill et al. [78] conducted a survey of about 100 network operators for their BGP routing policies, and reported that most ISPs follow the Gao-Rexford model in setting local preferences and exporting routes. Anwar et al. [79] identified cases where empirically observed routes violate either the Gao-Rexford model or the assumption of preferring shortest paths. They attribute these violations to causes arising from prefix-specific policies, hybrid and partial transit relationships, and geography-based policies. In this dissertation, I use measurements to reveal local preferences among providers of multi-homed ISPs, and to evaluate how often such preferences overwrites the shortest path assumptions. My results provide complementary aspects to Gao-Rexford

model. Furthermore, I evaluate the impact of such local preferences among providers on the performance of anycast: although unaware of the routing performance, these local preferences do not particularly increase anycast inflations.

Route manipulations, especially BGP poisoning, have been used as measurement methods to discover hidden network topology [80], to evaluate the prevalence of default routing [81] and to identify alternate back-up paths in the Internet [79]. Researchers have also proposed to use BGP poisoning as a mechanism to reroute traffic and avoid congestion links under DDoS attacks [82–84]. I augment route poisoning's ability to reveal local preferences with other mechanisms, including remote traffic engineering with community tags and selective AS-path (un-)prepending.

Chapter 3: Evaluating Anycast Performance

In this chapter, I aim to answer the following question: Does anycast actually achieve the desired properties (described in Chapter 1) that it seems to have in performance, scalability and reliability? [1] I describe work with colleagues which studies whether anycast is effective at achieving the properties through evaluation on the performance of many distinct anycast deployments. Also, the longitudinal measurements I collected provide insights on the benefit of adding anycast sites: how does anycast improves its performance as new sites were deployed? Further, I analyze the impact of routing instability on anycast, and present a heuristic on identifying the causes of routing changes that trigger significant performance shifts in anycast.

3.1 Dataset overview

I measured DNS root servers and major open resolvers to analyze the performance of Internet-wide anycast deployments.

[1] Evaluating and improving anycast resilience is not within the scope of this dissertation; however, we believe the dynamic hints described in Chapter 6 can be used to mitigate the effect of large-scale attacks like those that took place Nov. 30 and Dec. 1, 2015 [6, 85].

Root	Operator	Number of sites	Number of global sites
A	Verisign Inc.	15	15
B	ISI	2	2
C	Cogent Comm.	10	10
D	Univ. of Maryland	148	23
E	NASA (Ames)	196	95
F	ISC Inc.	187	107
G	US Dept. of Defense	6	6
H	US Army	2	2
I	Netnod	61	59
J	Verisign Inc.	123	71
K	RIPE NCC	62	61
L	ICANN	145	145
M	WIDE Project	5	4

Table 3.1: Root server overview, current as of April 2019.

3.1.1 DNS root servers overview

The DNS root service has been distributed over thirteen root servers, each referred to by a "letter", A-root through M-root. These root servers are assigned to thirteen Internet (IPv4) addresses, and operated by different entities. Over the past decades, all root servers have been *anycasted*, replicating DNS root service over hundreds of sites all over the globe. The same root address may be (and often is) announced

Operator	Respond to CHAOS	Number of sites
Google	No	28
OpenDNS	Yes	32
Cloudflare(Quad One)	Yes	174

Table 3.2: Popular DNS open resolvers overview, current as of April 2019.

through different ASes. In this dissertation, I consider each root to be an separate anycast deployment, and examine their performance independently. Table 3.1 lists the operators, and the number of global and local sites for each DNS root server in April 2019. Data in the table are from `http://root-servers.org`. In this dissertation, I study 9 out of 13 roots that have at least 5 anycast global sites. [2]

3.1.2 Major DNS resolvers overview

Many popular open DNS resolvers are anycasted. I focus on the anycast resolvers provided by Google [3], OpenDNS [86], Cloudflare [5]. They are the most popular open resolvers that are used by millions of people around the world everyday. These open resolvers replicate DNS recursive resolver service over hundreds of sites, and announce resolvers' addresses through different ASes. Table 3.2 lists the operators, and the number of sites for each of the major open resolvers as in April 2019. For these open resolvers, we consider all of their anycast sites are global.

[2] G-root does not respond to "hostname.bin" DNS CHAOS queries with meaningful identifiers that we use to distinguish sites. See section 3.1.3 for more details.

3.1.3 Passive and active measurements

Two different datasets are used in the analyses: traffic traces collected from sites of D-root server, and active measurements from RIPE Atlas probes. In the rest of this section, I describe these datasets and their features.

3.1.3.1 D-root server traffic traces

The first dataset is sampled traffic traces collected from all sites of D-root. D-root is operated by the University of Maryland. As of April, D-root had over 143 anycast sites, 23 of which were global and the rest local. Roughly 20% of all traffic at each site is collected. The analysis in this dissertation used longitudinal D-root traffic data collected on everyday from 2016 to 2018. On average, during these years, D-root received more than 30, 000 queries per second, resulting in about 140 GB of data per day.

This passive collection of DNS traffic provides us a global, detailed view of clients activity and query distribution seen at D-root. Also, this dataset shows traffic load distribution and variance among anycast sites, which allows us to evaluate the effectiveness of load balance in anycast.

However, this dataset represents only D-root, one out of 13 DNS roots. Since other DNS roots have different number of anycast sites and provider ASes, the performance evaluation based on this dataset does not immediately generalize to other anycast deployments. Also, this passively collected dataset does not provide insight into how the queries were directed to sites and route selection process. In

order to expand the analysis to other anycast deployments, and to better understand the interaction between anycast and BGP routing, I augment D-root dataset with active measurements conducted by RIPE Atlas platform.

3.1.3.2 RIPE Atlas measurements

The RIPE Atlas framework [87,88] contains ∼10,000 active probes distributed across 181 countries and in ∼3621 ASes as of April 2019. Each probe periodically executes pre-defined measurements, referred to "build-in measurements", that include specific DNS queries, pings and traceroutes to all 13 DNS root servers. The analysis focuses on two specific "build-in measurements": DNS CHAOS queries and traceroutes.

We refer DNS CHAOS query to the approach supported by BIND implementation of the DNS protocol suite to identify a particular server. Specifically, a DNS query for a TXT record in Class CHAOS (as opposed to the common case, Class Internet) for the domain name "hostname.bind." will return a unique identifier for the responding server, which is configured by the name server operator. For example, a typical response to DNS CHAOS query from D-root is "mcva2.droot". The record in this response indicates that the responding server is D-root replica #2 located in McLean, Virginia. F-root replica that returns responses with identifier "yyz1f.f.root-servers.org" is the F-root replica '1f' located in Toronto, Canada. 12 out of 13 DNS root servers are configured to provide a unique identifier for each replica: G-root does not respond meaningful identifiers that distinguish sites. Among the 12 roots configured with meaningful identifiers, I analyze measurements that the RIPE Atlas

probes sent to 9 of them that have at least five global anycast sites. Each active RIPE Atlas probe sends DNS CHAOS queries to each DNS root server's anycast address, every 4 minutes. These measurements enables the analysis that tracks which particular replica each probe is directed to by BGP routing at a given time during the three years.

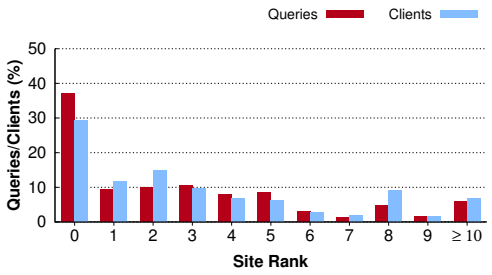
Along with DNS CHAOS queries, RIPE Atlas probes also send a traceroute to each DNS root's anycast address every 30 minutes. This traceroute measurement allows us to map the AS paths taken by queries from the probes to DNS root servers.

In this dissertation, I analyzed these measurements collected by all probes in about 4 years (March 2015 to March 2019). Prior work [8, 54] has used the same measurements to evaluate latency and client affinity. In my analysis, I augment these datasets with novel measurement methodologies that evaluate possible alternate routes towards different sites (in Chapter 4).

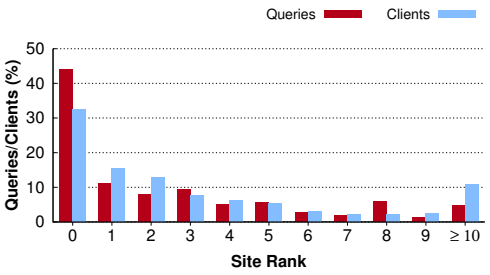
3.2 How does anycast perform for D-root?

In this section, I analyze the sampled traffic traces collected from D-root sites, and evaluate the performance of D-root anycast deployment.

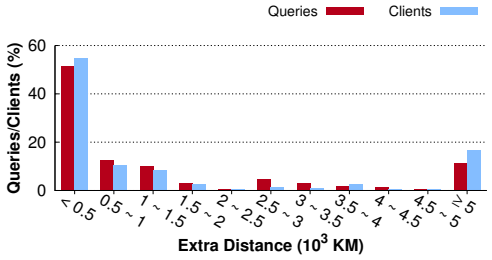
Figure 3.1a and 3.1a shows what fraction of queries or clients are directed to anycast sites ranked by distance to the sources, in July 2018 and March 2019. For each query received at D-root, I geo-locate the source address of the query using the MaxMind database [89]. Then, I compute the distance from the query source to the all D-root sites for each query (per query). For each query, the closest anycast site



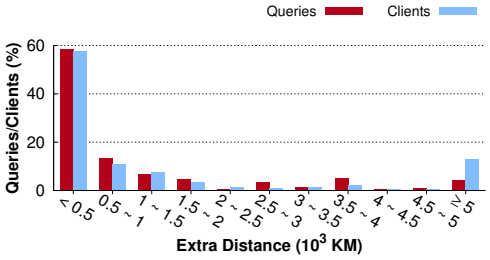
(a) Distribution of D-root queries by rank of the site used, as in July 2018.



(b) Distribution of D-root queries by rank of the site used, as in March 2019.

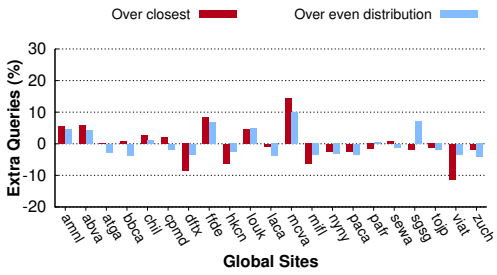


(c) Distribution of D-root queries by extra distance traveled, as in July 2018.

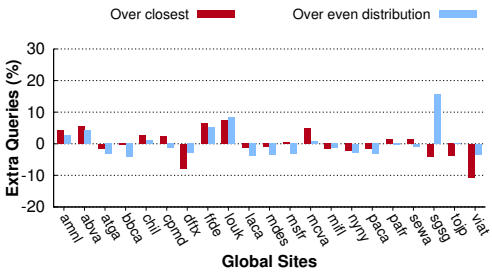


(d) Distribution of D-root queries by extra distance traveled, as in March 2019.

Figure 3.1: D-root performance based on sampled traffic traces



(a) D-root load in July 2018.



(b) D-root load in March 2019.

Figure 3.2: D-root load distribution

is ranked as 0, the second closest as 1, and so on. I compute the same measure for each source IP address (per client) as well. Figure 3.1a and 3.1b shows that only about 40 % of queries or clients to D-root were directed to geographically closest

site; About 10 % to the second closest site. Another 30 % of all queries or clients were directed to sites ranked 5 or higher.

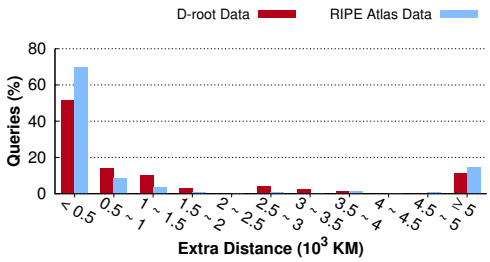
While Figure 3.1a and 3.1b reports that about 2/3 of all queries or clients are somehow "misdirected" by routing protocols to non-closest anycast site, it is possible that the higher rank sites are close to the clients as well. I evaluate the cost of inefficiency by measuring the extra distance that queries traveled to reach their anycast sites, over the geo-closest ones. Figure 3.1c and 3.1d shows that about 1/3rd of the queries traveled over 1000km more than minimal, and around 10.0% traveled more 5000km extra.

Geographic distance has been proven to be a reasonable approximation of expected latency [90]. The traffic dataset collected at D-root sites does not provide a direct measure of query latency. For various reasons, the geographically closest site may not be the site with lowest latency. In Chapter 4 and Chapter 6, I will quantify the anycast inefficiency and how much it can be improved not just for geographic proximity, but for measured latency as well. However, that analysis is based on traceroute and DNS CHAOS measurements from RIPE Atlas probes instead of DNS traffic at D-root.

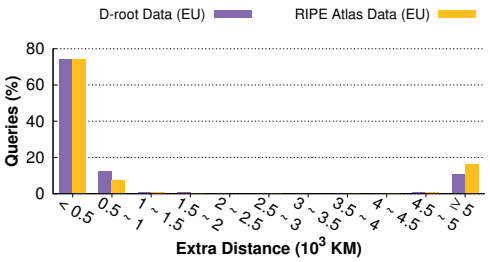
From these results, compiled across over years (2018 and 2019), from over 102 billion queries and 35 million IP addresses per year, representing over 190 countries. I conclude that there is substantial room for improving latency or geographic proximity performance in anycast for D-root. Maybe it is the case that the ineffective geographic proximity in anycast is to provide balanced query distribution among sites. Next, I characterize traffic load distribution on D-root sites.

Figure 3.2 shows the measures of load balance on D-root. The x-axis lists global sites of D-root; y-axis represents the fraction of total queries. Two different measures of load balance are considered, derived by comparing actual load distribution to two scenarios in where load is reasonably balanced. One scenario is the even load distribution, i.e., each site receives an equal amount of queries. The "Over even distribution" bars show fraction of queries, over (or under) the even distribution received by each site. In July 2018, for instance, Figure 3.2a shows that the *mcva* site received 10.0% of total queries more than its "fair share", whereas the *dftx* received 3.9% less. The other is the scenario when all queries were directed to their geographically closet site. The "Over closest" bars show the difference between such query distribution and the actual distribution. In July 2018, we see that *mcva* received as much as 14.4% of total queries more than it would have, if all queries had been directed to their closest site; *amnl*, *cpmd* and *ffde* also each received extra queries as much as 8% of total queries. During the same time, however, *viat* received 11.3% fewer queries. Figure 3.2b shows that queries distribution is still not balanced. Figure 3.2a and 3.2b show that query distribution among D-root sites is out of balance by different measures at different times.

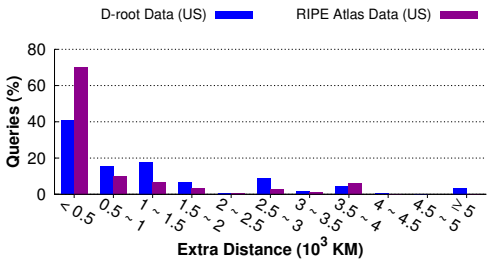
The above results show that anycast performs poorly for D-root: it is neither effective at directing queries to nearby sites, nor does it distribute traffic in a balanced manner. Is it that anycast does not perform well only for D-root, or these tends generalize to other anycast deployments? In the next section, I utilize RIPE Atlas measurements to study the performance of anycast for other DNS root servers.



(a) Comparison of all clients in D-root data and all RIPE-Atlas probes.



(b) Comparison of clients in D-root data and RIPE Atlas probes from EU-only.



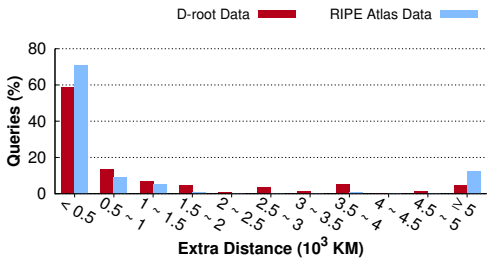
(c) Comparison of clients in D-root data and RIPE Atlas probes from US-only.

Figure 3.3: D-root clients vs. RIPE-Atlas probes: Extra distance traveled in July 2018

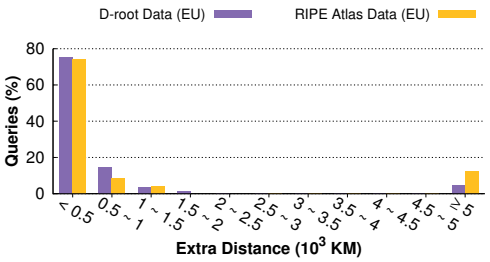
3.3 Are RIPE Atlas measurements biased?

The D-root traffic traces provide a global, unbiased sample of DNS clients distribution over the world, and their query volume. Unfortunately, I do not have access to traffic traces collected on other DNS roots or anycast services. Instead, I analyze active measurements conducted from RIPE Atlas probes to evaluate the performance of anycast for other DNS roots.

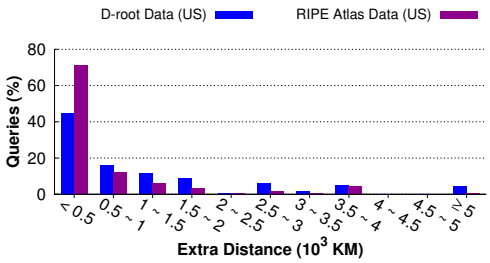
In contrast to D-root data, RIPE Atlas data present a partial view of anycast



(a) Comparison of all clients in D-root data and all RIPE-Atlas probes.



(b) Comparison of clients in D-root data and RIPE Atlas probes from EU-only.



(c) Comparison of clients in D-root data and RIPE Atlas probes from US-only.

Figure 3.4: D-root clients vs. RIPE-Atlas probes: Extra distance traveled in March 2019.

services measured from about 10,000 probes. The location of RIPE Atlas probes is publicly available. However they are biased towards Europe and the United States [91]. Hence, before the results based on the RIPE Atlas measurements are generalized, the bias of the data should be evaluated. The D-root data serve as a "ground truth" for how queries are distributed across anycast sites, and I evaluate the bias by comparing results based on RIPE Atlas data with the results derived from the D-root data.

Specifically, I measure the extra distance distribution (as in Figure 3.1c and 3.1d)

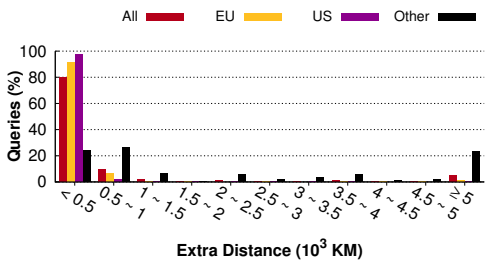
for queries from RIPE Atlas probes to D-root, and compare it with the result from D-root data. For RIPE Atlas probes, their locations are publicly available [91], and I identify the D-root sites they were directed to with the DNS CHAOS queries.

Figure 3.3 and 3.4 shows data over one week from both RIPE Atlas probes and D-root traces in 2018 and 2019 respectively. Due to the concentration of RIPE Atlas probes in Europe and United States, in addition to the overall result, Figure 3.3b and 3.4c plot the results specifically for queries from these two regions.

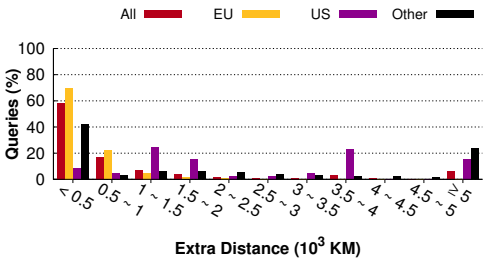
From Figure 3.3 and 3.4 we see that the results from RIPE Atlas measurements do not correspond well with the "ground truth" distribution based on D-root data. This shows that the results from RIPE Atlas measurements do not correspond well with the "ground truth" distribution derived from D-root data. The RIPE Atlas probes do not represent global clients distribution for D-root, especially outside of Europe. However, it is worth noting that in all cases, the results from RIPE Atlas probes *overestimate* the performance of anycast for D-root. Since the global clients distribution and activities for D-root should be similar to that of other DNS roots, I believe it is reasonable to evaluate the performance of anycast for different DNS roots. The performance in reality should be *worse* than results evaluated from RIPE Atlas measurements, as illustrated by the evaluation on D-root.

3.4 How does anycast perform for other roots?

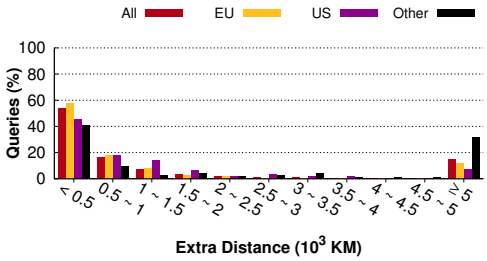
In this section, I show in Figure 3.5 and 3.6 the extra distance measure for three DNS roots in July 2018 and March 2019: C-, K- and L-root. The results for other



(a) C-root



(b) K-root

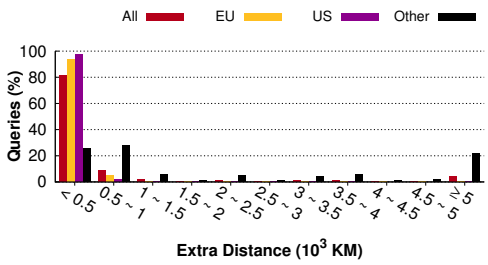


(c) L-root

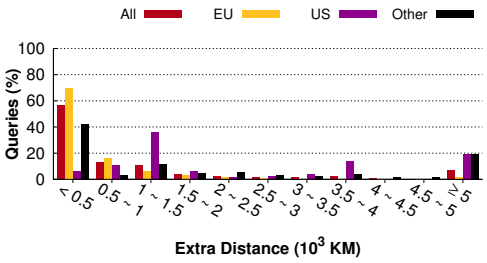
Figure 3.5: Distribution of RIPE-Atlas queries over extra distance (compared to their closest sites) traveled in July 2018.

roots are shown in Figure 8.1 and 8.2 in Chapter 8. The three roots are different anycast deployments operated by various entities: as in March 2019, C-root has 10 global sites; K-root contains 61 global sites; L-root has the largest number, 145, global anycast sites. I focus on C-, K- and L-root, along with D-root, because they are the ones with good unicast representative addresses for their sites, as shown in Section 4.2.

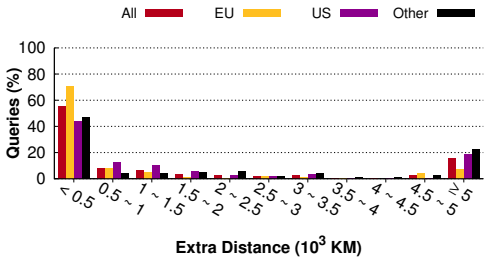
K- and L-root, operated by RIPE NCC. and ICANN, respectively, show performance similar to D-root. Also similar to D-root, the inefficient performance of K- and L-root persists in the evaluation 6 months later. C-root, which is operated by



(a) C-root



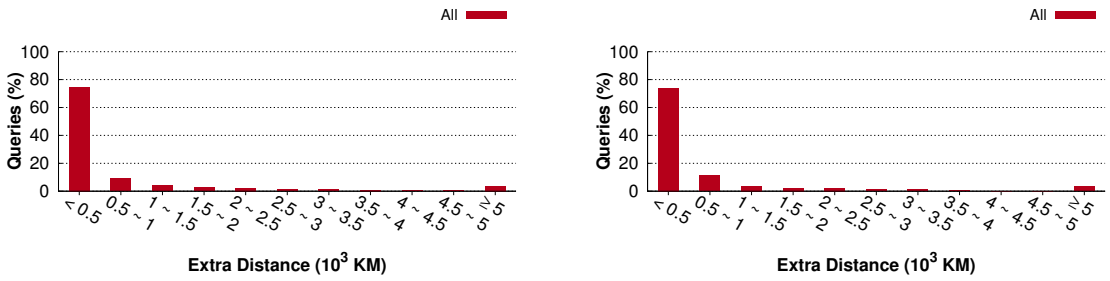
(b) K-root



(c) L-root

Figure 3.6: Distribution of RIPE-Atlas queries over extra distance (compared to their closest sites) traveled in March 2019.

Cogent, performs better than the other two, as well as D-root. It is expected that C-root performs well, since queries to C-root are largely directed to proper site by intra-domain routing. I have the following hypothesis to explain the better performance in C-root: The anycast service of C-root is provided through Cogent, a Tier-1 ISP with broad coverage and numerous peering points with other ISPs. Due to the vast usage of "early-exit" routing policy, most queries to C-root will be sent along a path that traverses providers without much of detour and enter Cogent at a nearby peering point. Once in Cogent's network, queries are direct to their sites based on intra-domain routing, which usually incurs little path inflation [92]. For C-root, it is



(a) In November 2018.

(b) In March 2019.

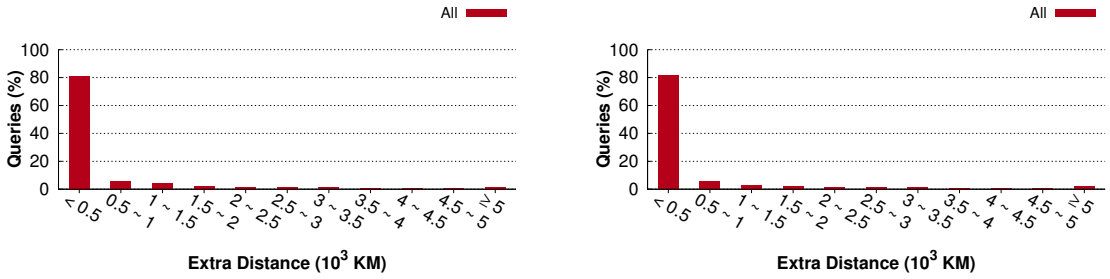
Figure 3.7: Distribution of RIPE-Atlas queries to Google open resolver over extra distance (compared to their closest sites) traveled.

unlikely that queries are directed to distant sites due to poor inter-domain routing selection, which is the main cause of inefficiency of anycast for other roots as shown in Chapter 4.

These results derived from RIPE Atlas measurements suggest that the performance deficit experienced by D-root is not special, but indeed representative of current anycast deployments. In Chapter 4, I describe novel techniques to investigate the causes for such inefficiency in anycast.

3.5 How does anycast perform on open DNS resolvers?

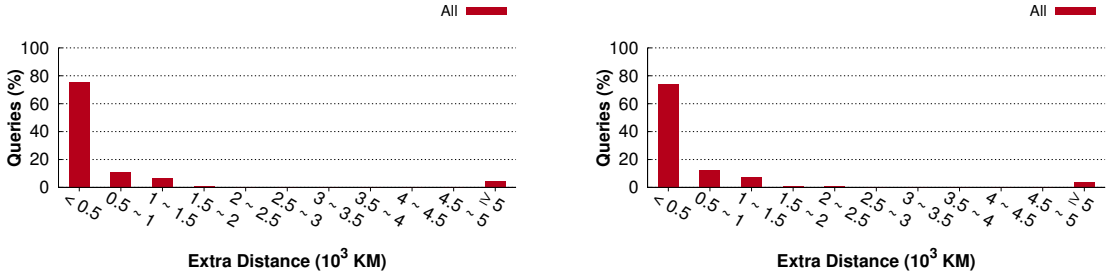
Anycast is becoming more and more popular in the Internet. Other than DNS roots, major open DNS resolvers are also built upon anycast. For example, open recursive resolvers hosted by Google (i.e., 8.8.8.8, 8.8.4.4), OpenDNS (i.e., 208.67.220.220, 208.67.220.222) and Cloudflare (i.e., 1.1.1.1, 1.0.0.1), are all anycasted. Unlike root DNS servers, these popular open DNS resolvers are frequently queried by millions



(a) In November 2018.

(b) In March 2019.

Figure 3.8: Distribution of RIPE-Atlas queries to Cloudflare open resolver over extra distance (compared to their closest sites) traveled.



(a) In November 2018.

(b) In March 2019.

Figure 3.9: Distribution of RIPE-Atlas queries to OpenDNS open resolver over extra distance (compared to their closest sites) traveled.

of users. Since DNS is involved in most daily activities of normal Internet users, the performance of the anycast deployments underlies these open resolvers is critical to almost all Internet users.

In this section, I evaluate the performance of anycast deployments under those open resolvers, and analyze if they experience similar problems as the DNS roots do. To conduct similar evaluations as in the last section, there are two requirements to the target anycast deployments:

- I need to be able to Identify the anycast site each query is sent to. For example, I use CHAOS queries to identify anycast site in DNS root servers.

- I need to know the locations of all sites in the anycast deployment.

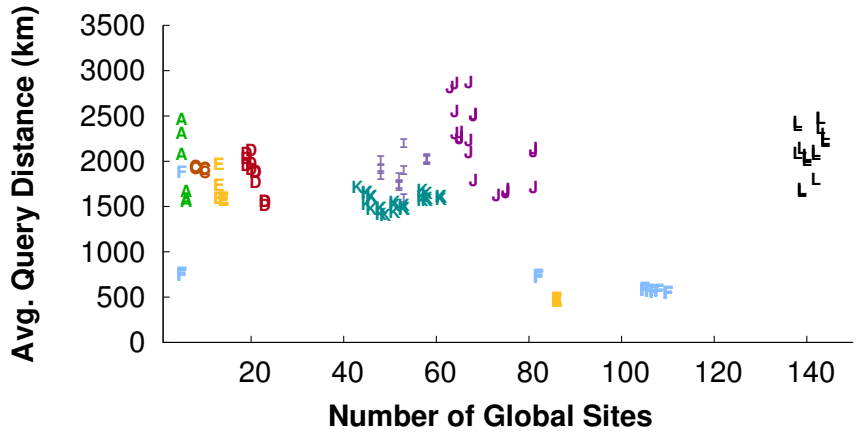
Only with these requirements, I can evaluate how far away the queries are sent to, and whether nearby sites are present.

Fortunately, three major open DNS resolvers satisfy both requirements: Google, OpenDNS and Cloudflare's Quad One. Both OpenDNS and Cloudflare respond to DNS CHAOS queries, from which I obtain identifiers that can locate the responding anycast sites. Meanwhile, these two open resolvers publish the locations of all their anycast sites [4, 5]. Googles open resolvers do not support CHAOS query (or similar queries). But Google publishes the unicast IP address ranges for its open resolvers to forward queries to authoritative DNS servers, and associated locations [3]. From the authoritative server's side, it identifies which anycast sites send the queries by source addresses. Based on this property of Google open resolver, I design the following technique to identify which site receives each query. I set up an authoritative DNS server for a domain that we control: "scriptroute.org". This server responds queries for random sub-domains of "scriptroute.org" with the source addresses of the queries. For example, when the server receives a query for "<random_string>.scriptroute.org" from address a.b.c.d, it will respond with a.b.c.d in the answer section. When I send queries for "<random_string>.scriptroute.org" through Google open resolver, the response will be the unicast address that used to forward my queries to "scriptroute.org" authoritative server. I can then identify

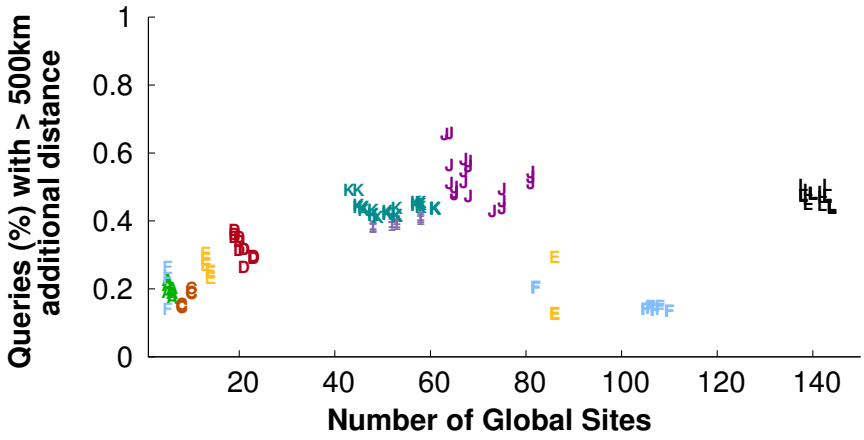
which site of Google resolver receive my query by mapping the unicast address in the response to its associated location. To increase the chance that open resolver forwards queries to the authoritative server, I set the TTL value in the response to be one second. Table 3.2 lists the open resolvers that I measure and their features.

Figure 3.7, 3.8 and 3.9 show the extra distance measure for the three open resolvers in November 2018 and in March 2019. All measurements are performed from RIPE Atlas probes. To balance measurement coverage and cost, I select one RIPE-Atlas probe from each (AS, country) tuple: about 3,100 probes are used in the measurements. These results provide a measure of the cost of misdirection, by quantifying the extra distance queries that are not directed to the closest site must travel. In Figure 3.7, over 20% of the queries to Google resolvers travel over 1000 km more than minimal, and over 8.0% travel more than 5000 km extra. The performance remains similar in the two measurements. I observe similar performance for OpenDNS resolvers, as shown in Figure 3.9. Anycast performance for Google and OpenDNS resolvers are slightly better than D-root, but there are still more 15% of queries directed more than 1,000 KMs away from the closest sites. Cloudflare open resolver seems to be more efficient than Google and OpenDNS. Figure 3.8 shows that bout 90% of queries are directed to nearby sites (e.g., less 1,000 KMs extra distance).

Although these open resolvers appear to have better anycast performance than the DNS root servers, there is room to improve. Consider the number of clients these resolvers serve, improving anycast performance for even 10% of the clients to the open resolvers could benefit millions of Internet users. In the next section, I use



(a) Average distance from probes versus number of global sites.



(b) Fraction of RIPE Atlas probes directed to a site within 500km of the geographically nearest.

Figure 3.10: How the number of global anycast sites affects performance.

longitudinal measurements to see if anycast benefits from adding more sites.

3.6 Does adding more anycast sites improve performance?

By tracking the number of anycast sites for each root, and analyzing RIPE Atlas measurements from 2017 to 2019, I obtain insights on how anycast improves its

performance as new sites were added. (Note that an existing site may add replicas, but that is not considered in the analysis.)

Figure 3.10 shows the performance of anycast versus the number of global sites for different root servers. x-axis is the number of global sites. For each measured root server, I evaluate the performance over each week from January 2017 to March 2019, and count the number of global sites in that week. Each measurement for a root from a week is represented as a point in Figure 3.10. Therefore, there are 115 points for each root (identified by the root letter and unique color in the plot): for example, over the measurement period, F-root increased from 5 sites to 110 sites. For each root, at each x-axis value (number of global sites), I show at most 4 performance values, including the 20th and 80th percentile among the values. I sample the points for legibility, while illustrating the variance in performance.

I use two different performance measures as y-axis. Figure 3.10a shows the average distance traveled by queries from RIPE Atlas probes to each root. This metric is an absolute measure of performance, and it is expected to decrease as the number of (global) sites increases. Figure 3.10b shows the fraction of queries that traveled more than 500km farther than to their closest global site. The extra distance traveled is a *relative* measure of performance, since the extra distance depends on the number and distribution of available sites. Thus the result in Figure 3.10b shows not only the performance of anycast, but more importantly, how efficiently new global sites are utilized.

For some root servers (including C-, D-, J- and L-root), the number of global sites is relatively stable over the year, and the vertical displacement of the letters

represent the variance of performance. In the next section, I will investigate into the effects of routing dynamic on performance of anycast. For other roots (including E-, F-, and K-root) placed many (e.g., 105 for F-root) global sites during the years. The results characterize the effect of such investment in anycast infrastructure. Unfortunately, even though F-root added 105 sites, its performance did not improve significantly, both in absolute and relative terms. In general, performance, somewhat counter-intuitively, is seemingly insular to the number of sites added.

3.7 Performance changes expose routing dynamics

The previous results show long-term persistent inefficiency in the performance of anycast for most DNS root servers and three open resolvers. In addition, the wide variation in measured performance under the same anycast deployment (e.g., the same number of sites, with the same providers.) is observed in Figure 3.10, especially for the root servers with a stable number of sites. Prior work [40–44] has shown that routing changes, especially inter-domain routing changes, directly influence data plane performance in unicast. In anycast, it is the Internet routing protocols that determine which site serves a client. One should expect the dynamic nature of the Internet routing could also cause fluctuation of end-to-end performance in anycast. In this section, I demonstrate how to use longitudinal RIPE Atlas measurements to (a) identify significant performance changes in anycast of DNS root servers; (2) develop heuristic to expose correspondent routing dynamics.

Provider	**Code**	**ASNs**
AT&T	AT&T	7018
Cogent Communications	COGENT	174
Deutsche Telekom AG	DTAG	3320
Global Telecom & Technology	GTT	3257, 4436
KPN	KPN	286
Level 3 Communications	LEVEL3	3356, 3549
Liberty Global	LGI	6830
MCI Communications	UUNET	701, 702, 703
NTT Communications	NTT	2914
Orange S.A.	OPENTRANSIT	5511
Qwest Communications	QWEST	209
Sprint	SPRINTLINK	1239
TATA Communications	TATA	6453
Telecom Italia	SEABONE	6762
Telefonica Network	TELEFONICA	12956
Telia Carrier	TELIANET	1299
XO Communications	XO	2828
Zayo Group	ZAYO	6461

Table 3.3: List of Tier-1 ISPs. The "Code" column lists the string by which the ISP is identified in our results.

3.7.1 Identify performance changes

I use the average distance traveled by queries from RIPE Atlas probes to anycast address as an indicator for performance. During most of the time, the average distance between RIPE probes and their chosen sites was not consistent. Fluctuations in average distance suggest some probes are routed to different anycast sites. And impulsive shifts in this measure show substantial routing changes that affect a large number of probes. By identifying changes in average distance, I can thus identify when did routing changes happen. Figure 3.12a, 3.13a, 3.14a and 3.15a shows the average distance that queries traveled in 2016 and 2017 from RIPE Atlas probes to different DNS roots. At a high level, these results show that average query distances remain relatively stable for months, but show sudden impulsive behavior that can affect average query distances by thousands of kilometers.

3.7.2 Characterizing Tier-1s' interaction with anycast

In this section, I describe how to character interactions between ISPs and anycast deployments. Once I observed a performance shift in anycast deployment according to the average distance measure, I would like to investigate the causes for it. I focus on the events that cause impulsive shifts in performance, since such shifts are usually results from a large number of probe start to send queries to new anycast sites that are thousands of kilometers further than their the old ones. The paths chosen by the ISPs that carried this traffic must have changed. Moreover, the route changes are probably related to large transit providers based on the large number of

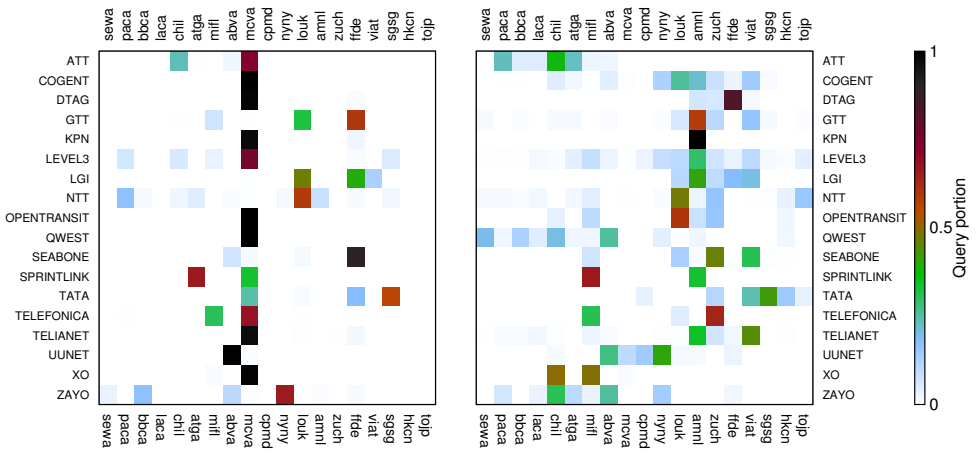


Figure 3.11: Distribution of queries routed by Tier-1 ISPs for D-root on Oct.1st 2016. The left panels shows which sites the queries went to; the right panel shows which sites are nearest to the RIPE Atlas probes.

probes affected and the significant distance between the new and old anycast sites. To investigate this hypothesis, I focus on how queries that are routed through Tier-1 ISPs (identified from the data in [93] and listed in Table 3.3) reach global anycast sites.

Note that I have two restrictions on the analysis. First, I only analyze queries routed through Tier-1 ISPs. Tier-1 ISPs have a global presence and many hundreds of peerings. Routing changes related to Tier-1 ISPs are likely affect a large number of clients, and thus cause significant performance changes. Analyzing Tier-1s allows us to understand how large ISPs interact with the DNS anycast. Second, I consider only queries to global sites. This is because local sites are advertised within small range, usually in one AS-hop, accounting for less than 10% of total D-root queries. Also, queries to local sites are mostly adhered to the sites and rarely change routes or traverse a Tier-1 to reach farther sites.

For each RIPE Atlas probe that originates in or traverses a Tier-1 ISP, I record (1) the site the query is directed to, and (2) the closest global site the query *could* have been directed to. I determine the ASes traversed by the queries from the traceroutes measurements to the anycast sites, described in section 3.1.3. The mapping from traceroutes to AS paths is based on methods described in section 4.3. Since RIPE provides accurate probe locations and I can identify which site is chosen at the time from CHAOS queries, I can compute where is the best sites for probes.

I use one example to explain how do I analyze the interactions between Tier-1 ISPs and anycast. Figure 3.11 contains two heatmaps. At left is a heatmap of global site to which queries from RIPE Atlas probes to D-root *were* directed on Oct.1st 2016, grouped by Tier-1 ISPs traversed. (If a query traversed more than one Tier-1 ISPs, the path is classified by the *first* one.) Darker shades represent higher query volume and the figure shows that most Tier-1 ISPs sent a large fraction of their traffic to the *mcva* or *abva*. Although there are 20 global D-root sites (at the time), the dark vertical line in this figure shows that most traffic is concentrated predominantly on one site. Meanwhile, many sites go virtually unused.

The right side of Figure 3.11 shows how the queries should be distributed if each query had been directed to its closest site. The distribution at right is a rough approximation of the locations of RIPE Atlas probes hosted by networks that need a Tier-1 to reach D-root. This figure represents what IP anycast could *ideally* achieve, and it shows what anycast's performance *should* to be: a far more even distribution of load and more low-distance queries than what actually happens.

From the heatmap, I identify many examples of pathological path length in-

flation: Deutsche Telekom, KPN, and Telianet direct most of their queries that originate in Europe to the *mcva* site in Virginia, bypassing multiple European D-root sites in Frankfurt, Amsterdam, and London. Similarly, queries routed through Cogent, QWEST, Opentransit, UUNet and XO could benefit from being routed to closer sites, but generally get routed to *mcva*.

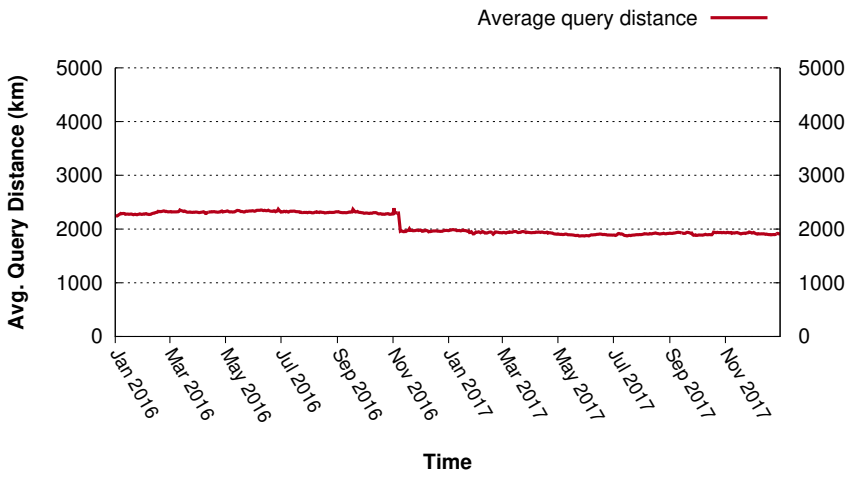
3.7.3 A heuristic to expose routing dynamics

In this section, I use the same methodology, i.e., analyzing the distribution of queries that traversed Tier-1 ISPs, to explain what exact routing changes caused significant performance shifts in average query distance. I use case studies to better demonstrate the heuristic. The cases are from several changes that affected C, D, E, and F-root in 2016 and 2017. Events for D, E, and F-root show clear changes in the AS paths: a set of Tier-1 ISPs changed how they reached the root address, typically choosing a single poor site. The C-root event does not reveal a change in AS path, but BGP advertisement traffic supports that a significant routing change was made.

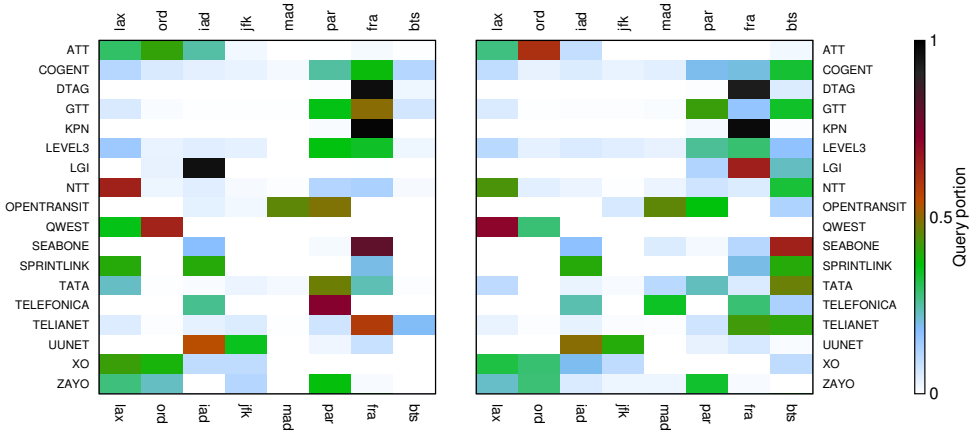
3.7.3.1 C-root: LGI chooses a better peering

In November 2016, Figure 3.12a shows that the average distance for queries to C-root decreased from 2300km to 2000km. (Because C-root is operated by Cogent, all queries traverse a Tier-1, meaning that the lines of the figure are overlapped; we use the difference to show the impact of routing changes beyond the Tier-1 ISPs.)

We next compare how Tier-1s routed queries the day before (Figure 3.12b) the



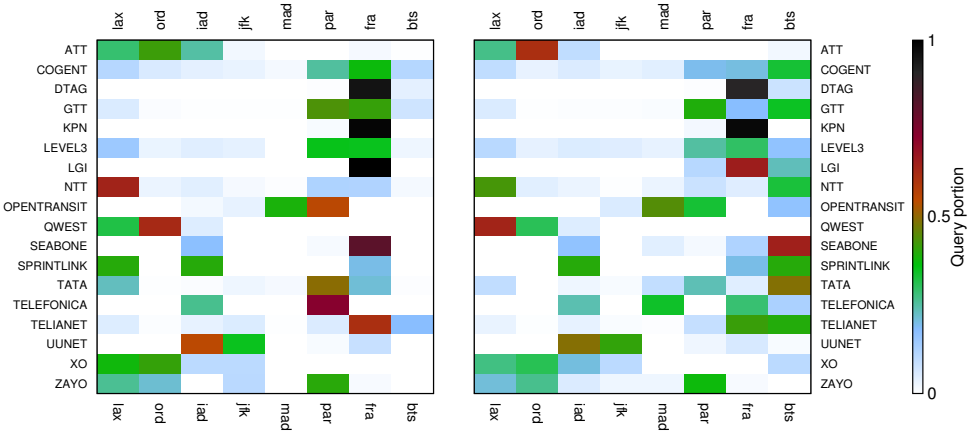
(a) Average query distance to C-root from RIPE-Atlas



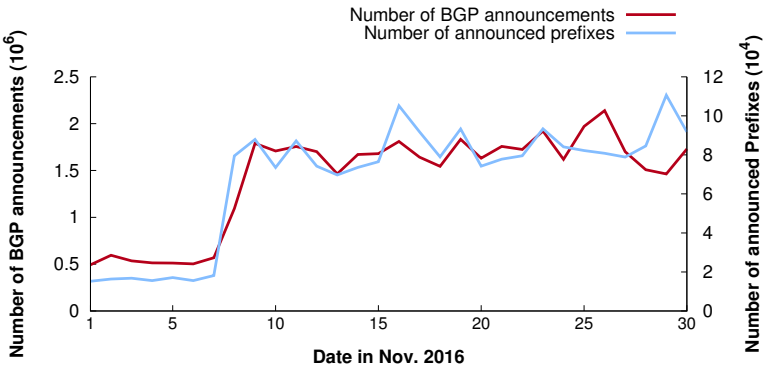
(b) Query distribution by Tier-1 on Nov. 7 2016

Figure 3.12: Distribution of queries to C-root routed by Tier-1 ISPs before and after routing change.

change and the day after (Figure 3.12c). As in Figures 3.11, the left shows where queries went and the right shows which site is nearest. The key difference between Figure 3.12b and 3.12c is that traffic from LGI is routed instead to Frankfurt (*fra*),



(c) Query distribution by Tier-1 on Nov. 9 2016



(d) Number of BGP announcements through LGI-Cogent peering, in early November 2016.

Figure 3.12: Distribution of queries to C-root routed by Tier-1 ISPs before and after routing change.(Cont'd)

nearer to the clients that it supports.

Because C-root is operated by Cogent, and LGI peers directly with Cogent, we sought to confirm that there was a significant routing change that occurred. In IP address space, the paths clearly traverse a different set of IP addresses to cross the peering. In BGP, the volume of BGP traffic associated with LGI to Cogent

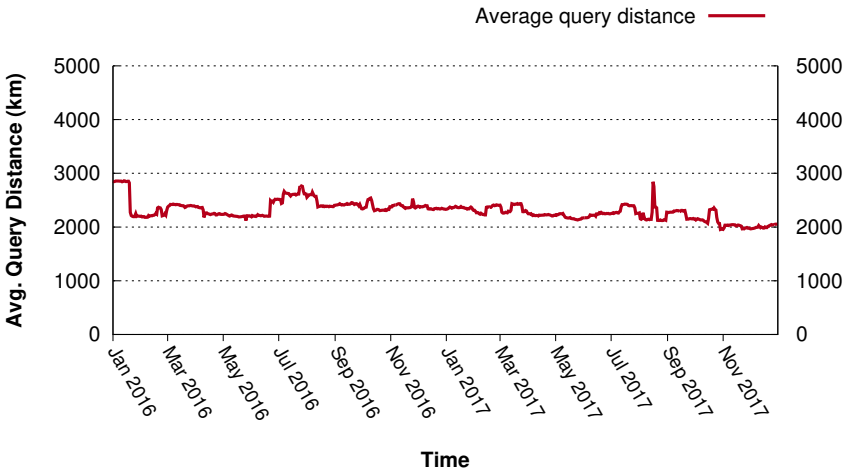
increased significantly at the same time, as shown in Figure 3.12d. This analysis uses BGPStream [94] to see BGP updates collected from RouteViews, focusing on prefixes advertised with the tuple LGI-Cogent (AS6830-AS174) in the AS Path. The plot shows that the number of announcements and prefixes with LGI-Cogent tripled around November 9, suggesting increased connectivity between the two.

3.7.3.2 D-root: Telia pulls DTAG to *mcva*

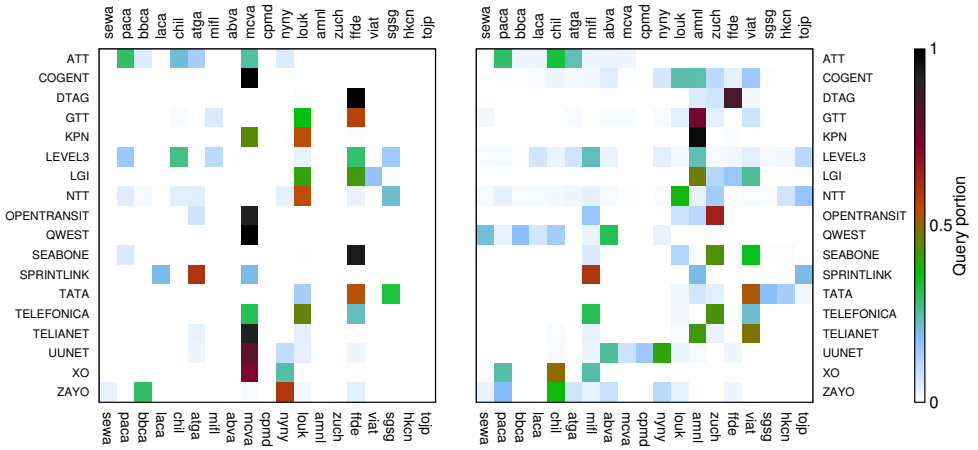
In June 2016, Figure 3.13a shows that the average distance for queries to D-root increased by about 300km, or by about 1000km if considering only queries that traversed Tier-1 ISPs. The key difference between Figure 3.13b and 3.13c is a shift toward the *mcva* site for DTAG.

Figures 3.13d, 3.14d, and 3.15d, described in more detail below, share a common dataset and format. The underlying dataset comprises traceroutes taken from RIPE Atlas probes to the root server's anycast address. Concurrently, RIPE probes query a special record from the root name server to determine which one was in use at the time. We translate the IP addresses of hops along the path into their originating AS to construct the traceroute-based AS path, then show only edges after the ISPs involved in route changes.

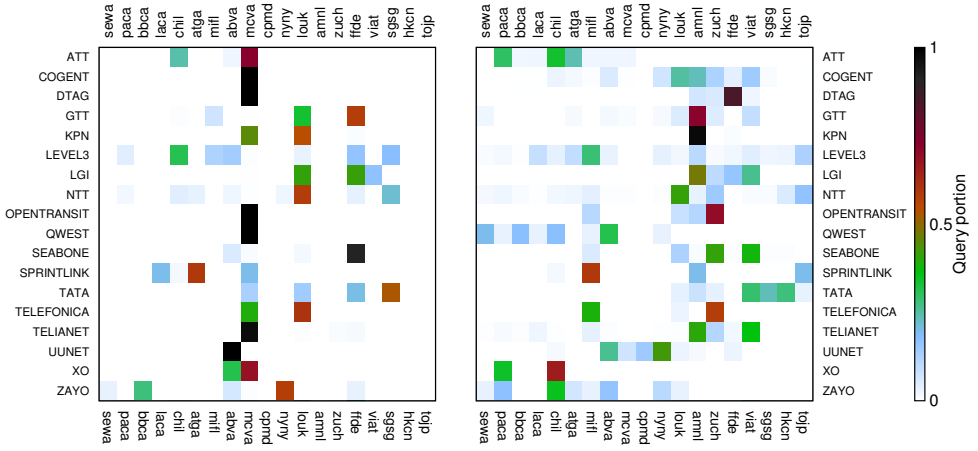
In the figures, numbered nodes indicate Tier-1 ASNs that were part of a routing change, or non-Tier-1 ASNs they used to reach a site. Named nodes at the bottom indicate the sites that were reached by this set of ISPs. In this set of changes, the number of sites in use is reduced through the change. Line style and thickness



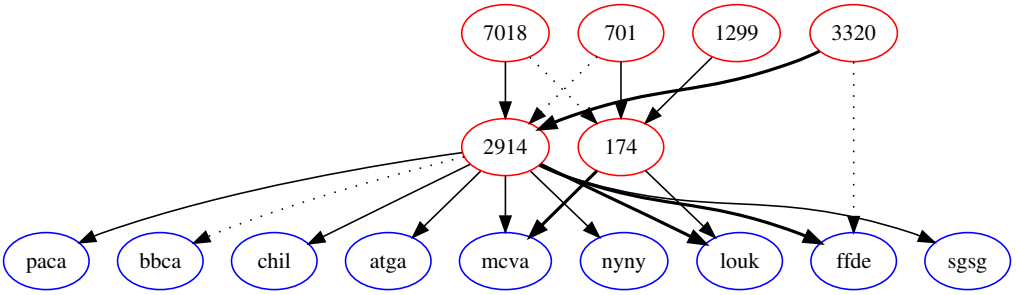
(a) Average queries distance to D-root from RIPE-Atlas.



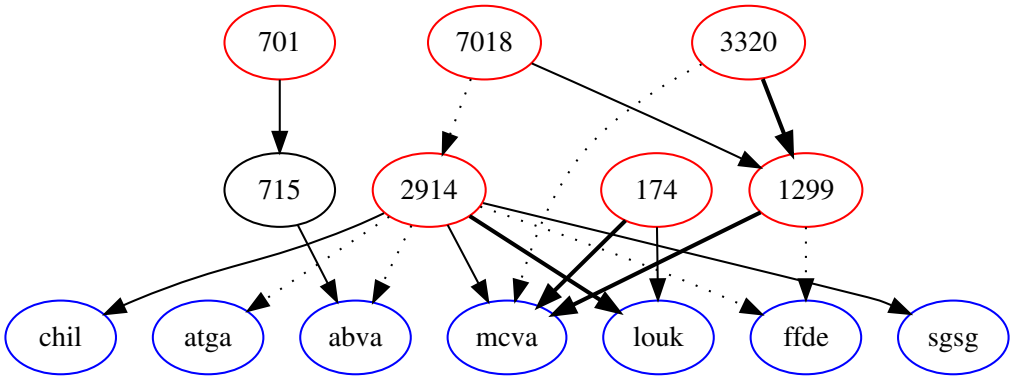
(b) D-root on Jun. 20 2016



(c) D-root on Jun. 25 2016



(d) AS paths on Jun. 20 2016



(e) AS paths on Jun. 25 2016

Figure 3.13: Distribution of queries to D-root routed by Tier-1 ISPs before and after the routing change. (a) Average query distance over time, (b) Query distribution by first Tier-1 ISP before and (c) after, (d) AS paths evident in traceroutes before and (e) after. In the AS graphs, edges represent appearance in traceroutes from at least 4 sources, solid edges at least 15, and thicker edges at least 100.

indicates the number of traceroutes that included a link from one AS to another. No edge appears if fewer than four traceroutes included such a link. Edges appear dotted unless seen at least 15 times: on one hand, relatively few observations may

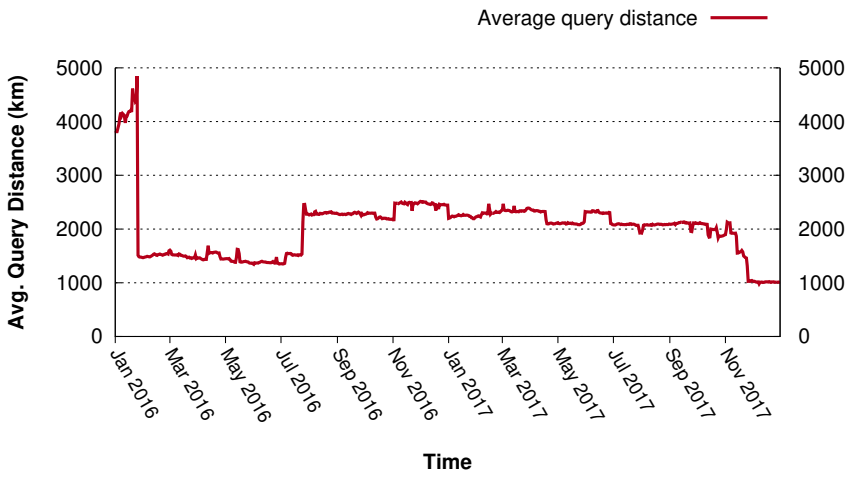
be due to transient behavior, on the other, omitting these edges may hide diversity that does not happen to be observed by RIPE probes. Plain edges are up to 100 observations, then lines are slightly thicker to 800, and in one case where roughly 1/10 of all RIPE probes used the connection from 3356 to 3549, a thickest line.

Figure 3.13d and 3.13e shows before and after Telia (1299) provided a direct route to the *mcva* (northern Virginia) site, rather than use Cogent (174). DTAG (3320) and AT&T (7018) switched routes to D-root from NTT (2914) to Telia (1299). Telia appears to direct most all traffic to the Northern Virginia (mcva) site, and did so even before the event when it first traversed Cogent (174) address space.

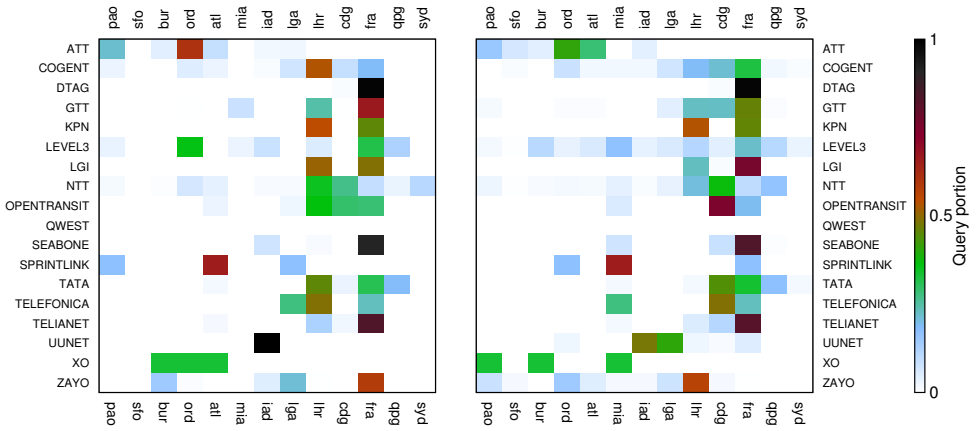
The precise scenario is unclear, but this event would reinforce that, to avoid sending its own traffic far, a Tier-1 should not peer with an anycast operator in just one location. In the event that a single peering is desired, to avoid collecting traffic to be sent far, a Tier-1 should avoid exporting a route to others when having a connection to only one site.

3.7.3.3 E-Root: 3356 starts advertising a route

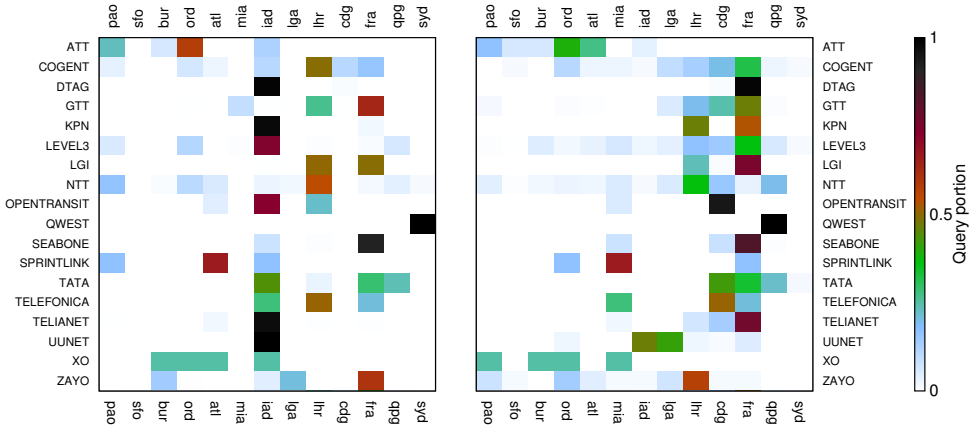
In July 2016, Level3 appears to have begun treating an AS it acquired (3549) as a sibling, re-advertising the route to E-root, instead of as a peer where it would not re-advertise. This general change in relationship between 3356 and 3549 has been documented by Dyn research [95]. The impact of this change appears in Figure 3.14a, increasing the distance from RIPE Atlas probes to E-root by 800km, and for the subset of queries that traversed a Tier-1, 1500km.



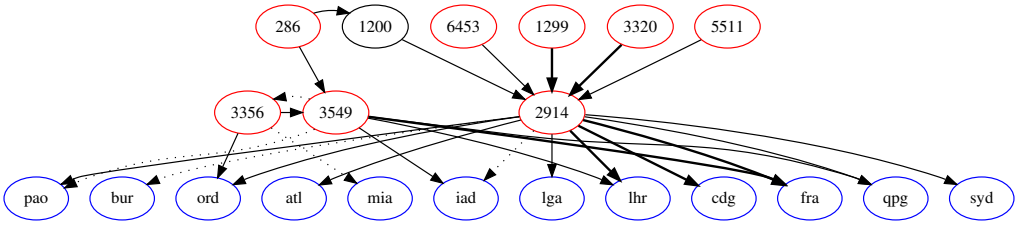
(a) Average query distance to E-root from RIPE-Atlas.



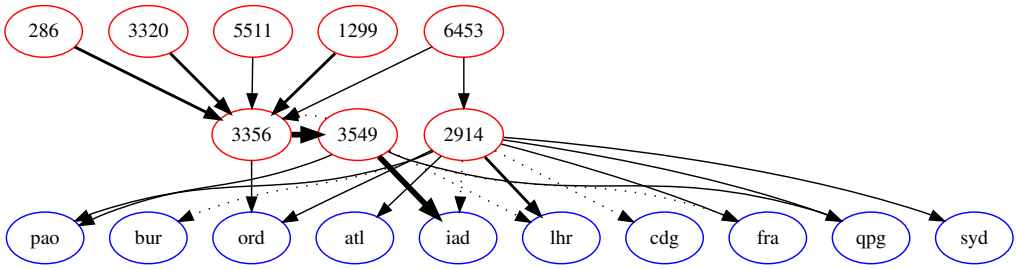
(b) E-root on July 24 2016



(c) E-root on July 26 2016



(d) AS paths on July 24 2016



(e) AS paths on July 26 2016

Figure 3.14: Distribution of queries to E-root routed by Tier-1 ISPs before and after the routing change. (a) Average query distance over time, (b) Query distribution by first Tier-1 ISP before and (c) after, (d) AS paths evident in traceroutes before and (e) after. In the AS graphs, edges represent appearance in traceroutes from at least 4 sources, solid edges at least 15, and thicker edges at least 100. Extra thickness represents the over 800 traceroutes that traversed the 3356 to 3549 and 3549 to iad site links.

Figure 3.14c shows that various providers switched from a site that was appropriate for the client set (typically Frankfurt/*fra*) to northern Virginia (*iad*). Figure 3.14e shows the change to the AS path involved. Various providers that previously used NTT (2914) to reach E-root chose the new Level3 route, although 3356 directed those queries to a specific address within 3549, which then sent those

queries to Northern Virginia (*iad*).

3.7.3.4 F-Root: Comcast advertises a route to Chicago

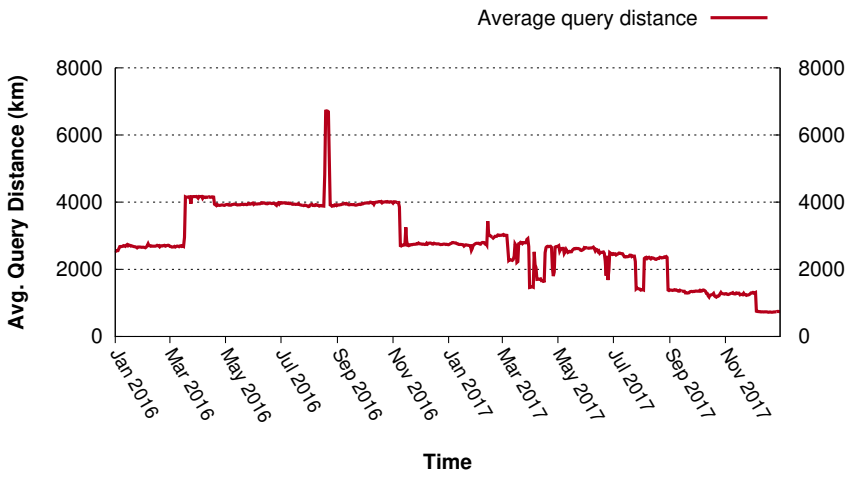
In March 2016, the average distance to F-root increased by almost 1,300km, as shown in Figure 3.15a. This is the result of shifting substantial traffic to the Chicago site, shown in Figures 3.14b and 3.14c.

Figures 3.15d and 3.15e show before and after Comcast (7922) appears to have advertised a route to F-root, despite delivering queries it received only to the Chicago (ORD) site. Notable is the prior diversity of sites (5 vs., in practice 1) and paths for this set of ISPs. 7922 may be seen as a customer by other ISPs, which could explain why so many Tier-1 ISPs chose the route to F-root through 7922.

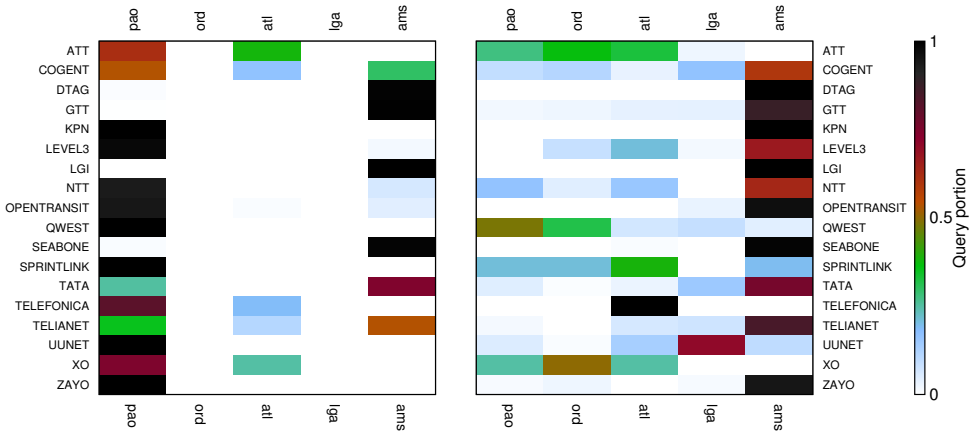
In this plot, the middle tier (7922, 2914, 1280, etc.) are only shown for traceroute paths that traverse the ISPs above. For example, the connection from UUNET (701) to Palo Alto (PAO) appears over 100 times overall in the data, but appears only rarely in a 12956-to-701-to-pao path. This change was corrected in November 2016, as can be seen in Figure 3.15a.

3.8 Conclusion

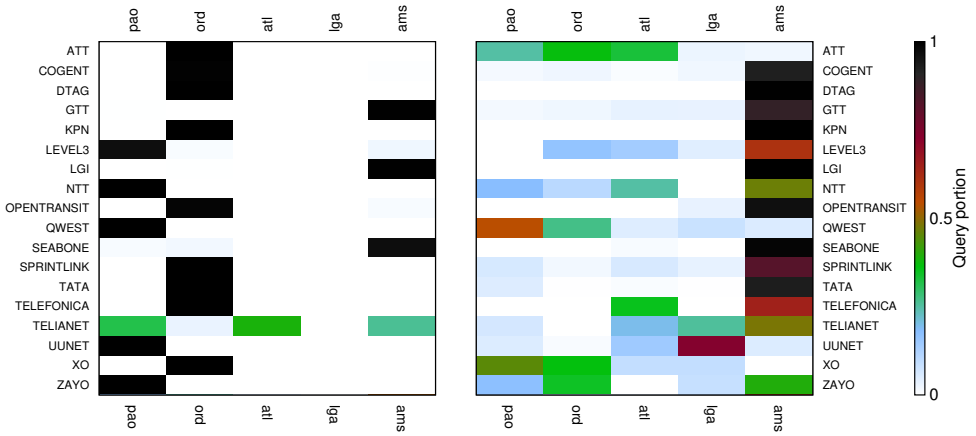
In this chapter, I show broad and longitudinal evaluation of distinct anycast deployments. I use passive traces collected from D-root sites over 3 years, as well as active measurements performed from over global distributed 8,000 RIPE Atlas probes, to understand how far do queries travel to anycast sites and how well are queries dis-



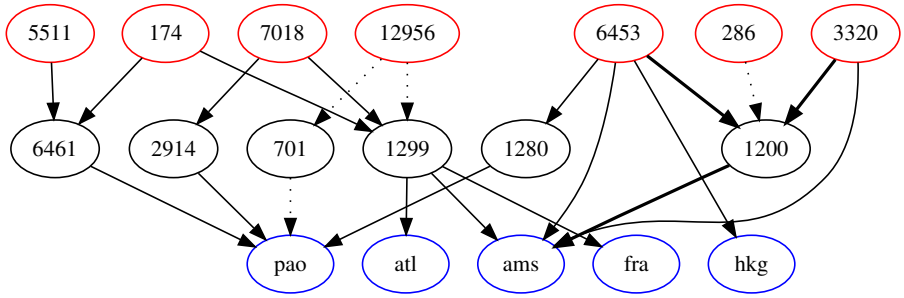
(a) Average queries distance to F-root from RIPE-Atlas.



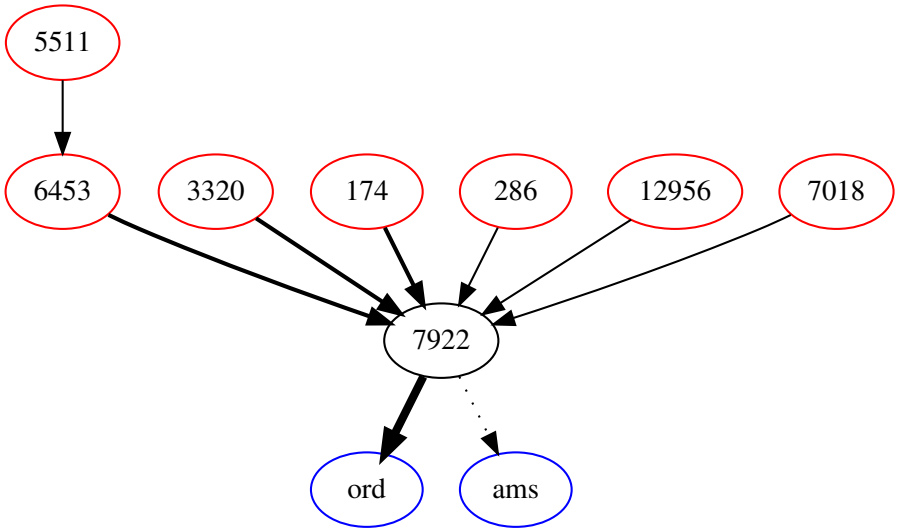
(b) F-root on March 15 2016



(c) F-root on March 20 2016



(d) AS paths on March 15 2016



(e) AS paths on March 20 2016

Figure 3.15: Distribution of queries to F-root routed by Tier-1 ISPs before and after the routing change. (a) Average query distance over time, (b) Query distribution by first Tier-1 ISP before and (c) after, (d) AS paths evident in traceroutes before and (e) after. In the AS graphs, edges represent appearance in traceroutes from at least 4 sources, solid edges at least 15, and thicker edges at least 100. Extra thickness represents the over 800 traceroutes that traversed the 7922 to ord site link.

tributed across them. The results show that about 30% of the queries traveled over 1000 km more than minimal; meanwhile, popular anycast sites attract over 10% of

total traffic more than its "fair share". By measuring nine DNS root servers with distinct anycast deployments (i.e., different number of global or local sites, different providers, etc.), I find that anycast is neither effective at directing queries to nearby sites, nor distributing load in a balanced manner in general. Not only DNS root servers, I also study anycast deployments in major DNS open resolvers, operated by Google, OpenDNS and Cloudflare. These open resolvers appear to have better performance than most DNS roots, however, there are still about 15% of the queries to them travel more than 1,000 km extra distance. With the longitudinal measurements of DNS roots over two years, I continuously track anycast performance overtime. Counter-intuitively, having more anycast sites seems to have little correlation to anycast performance. Further, I present a heuristic on identifying the causes of performance shifts in anycast with case studies. These cases demonstrate how a small change in routing decisions could cause significant change in anycast performance, and lead us to further investigate the root causes of performance problems in anycast.

Chapter 4: Diagnosing Anycast Performance Problems

In the previous chapter, I show that anycast, for most DNS root servers and major open resolvers, is ineffective at providing good geographic proximity for queries or balancing traffic load across sites. Packets are directed to certain anycast server based on BGP, the de facto interdomain routing protocol. However, BGP is a policy-based protocol and lacks mechanisms to pick the routes with better performance. Intuitively, BGP may create circuitous paths that are longer than the geographic distance between endpoints would require, which is characterized as path inflation in [92] Anycast allows BGP to select not only a circuitous path, but one that does not even lead to a nearby site. In other words, anycast introduces extra path inflation compared to unicast.

In this chapter, I quantify and compare the two sources of path inflation: unicast path inflation that is well known and studied, and anycast path inflation that is specific to routing in anycast. I develop novel measurement methodologies that allow me evaluate the performance of alternate anycast sites, and gain insights on how ISPs end up selecting the poor routes to distant anycast sites. Moreover, by conducting experiments, I am able to identify the causes of anycast path inflation and quantify how much each cause attributed to the overall inflation. I show that

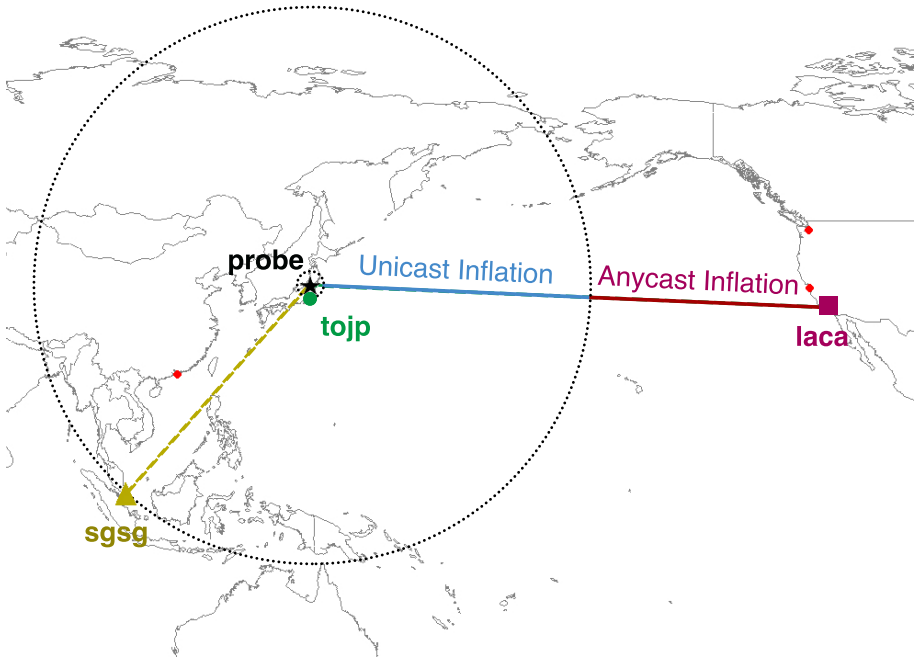


Figure 4.1: Illustration of anycast inflation compared to unicast inflation using a real example. The probe in Japan has no direct route to the closest site 'tojp' and was directed to 'laca', however 'sgsg' is the site that provides lower latency to the probe.

the majority of anycast path inflation is caused by unknown, sometimes arbitrary selection among seemingly "equal" routes by ISPs.

4.1 Anycast and unicast path inflation

In this section, I describe *unicast* and *anycast* path inflation. I refer to *unicast path inflation* as the path inflation expected from typical unicast routing caused by BGP policies and peering. Unicast routing is subject to path inflation in which the path taken is longer than necessary. Spring et al. [92] decomposed path inflation into topology and policy at the intra-domain, peering, and inter-domain levels, where

each layer could add to the path distance either by incomplete topology (the lack of a good path) or poor policy (choosing a poor path). Obviously, anycast routes will also be subject to similar inflation. In addition to *unicast* path inflation, I refer to *anycast* path inflation as the path inflation attributed specifically to routing of anycast addresses: the path inflation over the closest anycast site due to selecting routes to a distant site. These two sources of path inflation are illustrated in the following example.

Figure 4.1 is derived from a real example in RIPE Atlas measurements. This figure shows the case when a RIPE Atlas probe outside Tokyo, Japan, sends DNS queries to D-root. Queries from this probe are directed to the D-root site *laca* in Los Angeles, CA. The closest D-root site to this probe is the site *tojp* in Tokyo, Japan. However, it turns out that the D-root site *tojp* in Tokyo does not provide the lowest latency to the probe: using the measurement methodology described in 4.2, I find that the there is no direct route (that does not traverse the United States) from the probe to the site in Tokyo. Instead, the D-root site that provides the lowest latency is in Singapore *sgsg*.

In this example, the extra distance from Tokyo to Singapore can be considered inflation due to routing policy, thus attributed to *unicast path inflation*. However, the latency difference between the probe–Singapore versus probe–Los Angeles is due *anycast path inflation*. Anycast path inflation quantifies the extra performance deficit incurred by routing of anycast addresses for not selecting shorter routes that are available via unicast. In the rest of this section, I quantify anycast path inflation, and identify the underlying causes.

4.2 Unicast representatives of anycast sites

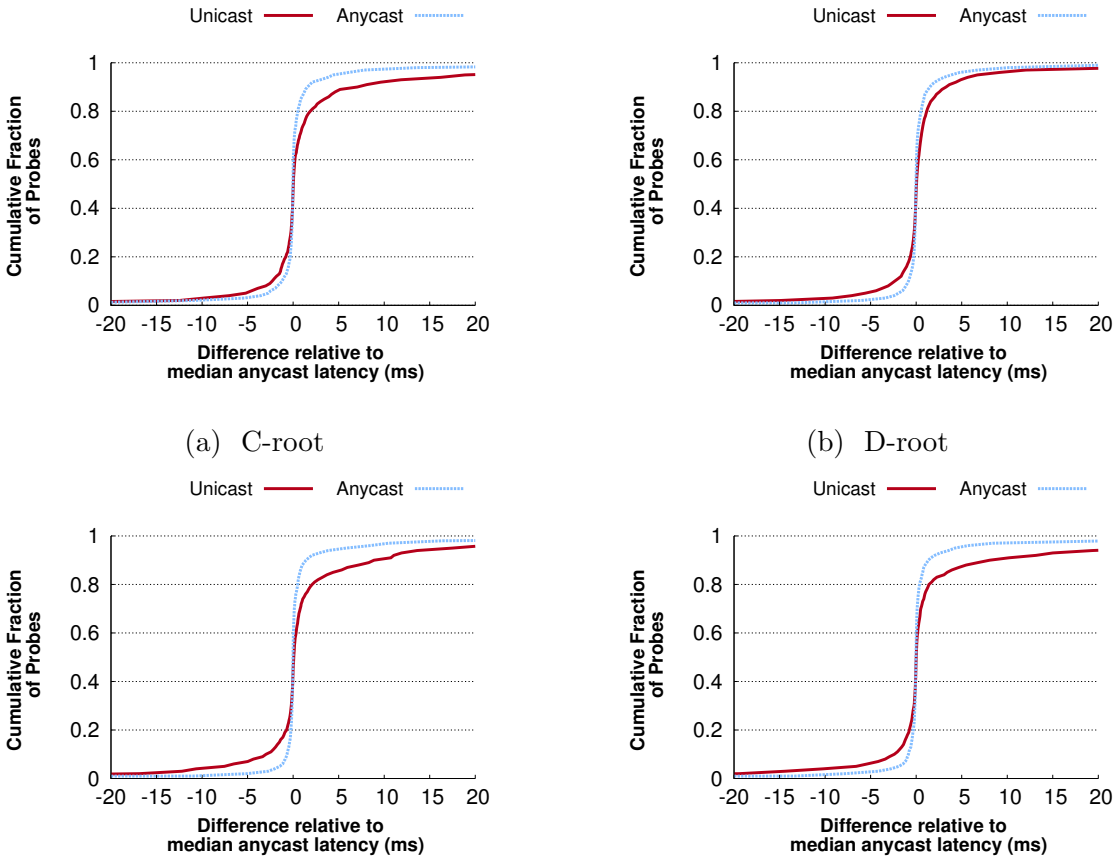
In order to quantify anycast path inflation experienced by a client, and compare it with unicast path inflation, I plan to evaluate the performance of alternate anycast sites that *could* have been chosen for the client. Note that packets sent to the anycast address will land on the site determined by BGP routing, and the performance of alternate sites remains undiscovered. In this section, I present the novel measurement methodology used to evaluate the performance of different anycast sites for a client.

4.2.1 Selecting unicast representatives

A unicast representative for an anycast site is a unicast address that is geographically close to the anycast site, and shares (substantially) the same network path when reached from a source that is directed to that site via anycast. Preferably, a unicast representative should be contained within the AS that advertises the anycast site.

For C-, K-, and L-root, they publish the unicast address used for management of individual sites.[1] I simply pick one address per site as the unicast representative address for that site. Although the management addresses have been used to evaluating anycast performance [7], ISPs may treat the management addresses differently from the anycast addresses. I evaluate if they serve as good representatives for anycast sites.

[1] F-root publishes management addresses too, but only for sites that are not hosted by Cloudflare.



(a) C-root

(b) D-root

(c) K-root

(d) L-root

Figure 4.2: Unicast representatives show latency performance similar to the anycast site they represent. The "Anycast" line shows the difference in latency between a single sample of anycast and the median, as a baseline for comparison. The darker line labeled "Unicast" shows the difference between a measurement of the unicast representative and median of anycast samples.

Other root DNS servers (e.g., D-root [96]) locate their servers at Internet Exchange Points (IXPs), the unicast representatives of their anycast sites are the representatives at corresponding IXPs. Packet Clearing House (PCH) operates route collectors at more than 150 IXPs, and releases the BGP routing tables collected from

C-Root Sites	% Agree	D-Root Sites	% Agree	K-Root Sites	% Agree
bts	90.7%	*abva*	96.2%	*at-vie*	69.0%
fra	91.8%	*amnl*	96.1%	*bg-sof*	86.2%
iad	92.9%	*chil*	97.3%	*ch-gva*	83.3%
jfk	91.7%	*ffde*	92.4%	*cl-scl*	52.3%
lax	91.8%	*hkcn*	80.0%	*de-ham*	96.4%
mad	85.9%	*louk*	95.5%	*es-bcn*	81.8%
ord	95.7%	*paca*	99.4%	*fr-par*	65.5%
par	81.4%	*tojp*	95.8%	*rs-beg*	73.3%
qro	100.0%	*viat*	96.6%	*us-ric*	70.8%
sin	96.5%	*zuch*	84.9%	*za-jnb*	70.0%

Table 4.1: AS path agreement between unicast representatives and sites; ten sites per letter are shown.

these route collectors [97]. These routing tables provide us with other (unicast) prefixes that directly connected at the IXP. I choose an address from the smallest unicast prefix at an IXP as the unicast representative of the collocated anycast site.[2] Preferably, the prefix should be from the AS that advertises the anycast site, e.g., PCH (AS42) for D-root.

With this heuristic method, I obtain unicast representatives for D-root global

[2]E-root also uses PCH and does not publish management addresses, but recently also started distributing via Cloudflare, making this technique of IXP-based representatives incomplete for E.

sites. Note that two global sites are missing *mcva* and *cpmd* since they are not collocated with IXPs. Fortunately, these two sites are disproportionately chosen by many queries originated from nearby or distant regions, so routes to *mcva*/*cpmd* are already shown in most cases.

4.2.2 Goodness of unicast representatives

Using the method just described, I select unicast representatives for C-, D-, K- and L-root. Before measuring alternate anycast sites using the unicast representatives, I evaluate how well they represent their corresponding anycast sites. From each vantage point, I measure the latency as well as the traceroute path to the anycast site through anycast address, and to the corresponding unicast representative through its unicast address. I compare the measured latencies and check if the paths overlap.

Recall that RIPE Atlas probes allow people to send DNS CHAOS queries and traceroutes, to both anycast and unicast representative addresses. Each probe (vantage point) provide me measurements that used to evaluate one single site (and the corresponding unicast representative) per root. From DNS CHAOS query measurements, I obtain which probe uses which anycast site. (I only select probes that have stable affinity to their sites during measurements) I assign probes to measure the unicast representative corresponding to the site it used, so a different number of probes may be used to measure different sites. Due to measurements budget, I use about 2,000 probes to measure their anycast sites and unicast representatives for each root. Probes are assigned across different sites, limiting to at most 200 probes

per site for C- and D-root, 30 probes per site for the larger K and L. Some sites will be assigned with fewer probes if too few probes are directed to that site through anycast. From each probe, I send traceroutes to both the anycast address and to the unicast representative of the site the probe used. I obtain both latencies and IP-level paths from the probe to the anycast address and to the unicast representative. Note that this is a one-time measurement.

Figure 4.2 characterizes the difference between latency to the unicast representatives and the latency to corresponding anycast sites of C-, D-, K- and L-root. To account for the natural variance in latency, I also obtain the median latency from the probe to anycast address (using RIPE Atlas' built-in ping measurements) during the one-hour window around the time I conduct measurements. I refer this median latency as median anycast latency from probe to anycast site. Then, I compute the differences of both one-time measured latencies (to anycast address and to unicast representative) to the median anycast latency. I aggregate latency differences from all selected probes and plot them as CDFs as shown in Figure 4.2. In the plots, x-axis is the latency difference relative to the median anycast latency; y-axis shows cumulative fraction of probes. 'Anycast' line shows the difference between individual anycast measurement and the median, which serves as a baseline; 'Unicast' line plots the difference between individual measurement to the unicast representative and the median anycast latency is a measure of representativeness: the closer 'Unicast' line is to the baseline, the better representativeness the unicast addresses have. For the measured roots, the comparison in Figure 4.2 shows that the unicast representatives are not routed in a way that systematically degrades (or

improves) their performance.

In addition to similarity in latency, the traceroute measurements allow me to evaluate the similarity in AS level paths to anycast sites versus unicast representatives. I use the method described below in 4.3 to infer AS level paths from traceroutes. Table 4.1 shows a sample of sites from different roots and the fraction of probes that have matching AS paths to the anycast sites and to corresponding unicast representative. The AS paths show a close match overall, with around 90% for C, 90% for D, 75% for K, and 85% for L-root of the probes have matching AS paths. The AS path matches for C- and D-root were better than for K- and L-root. One difference between the two is that C- and D-root have single hosting ASes (Cogent and PCH) from which unicast representatives are drawn, while K- and L-root have different hosting ASes at different sites. However, I do not expect complete agreement, since unicast and anycast addresses are in different prefixes that may be routed differently.

4.3 AS path inference

Section 4.2 presents a methodology to measure the performance of different anycast sites. In this section, I describe an AS-level path inference method that provides insights on how BGP routing ends up choosing poor routes to the anycast prefix. For each client, I compare its path to the chosen anycast site with the path to a unicast representative of a closer site, thus to locate the "decision point" where the two paths diverge. It is at this "decision point" that route selection failed: although

a good path exists to the closer site (i.e., the path to the unicast representative), a path to a different site was preferred. By identifying the "decision point" AS and its next-hop ASes in the routes to different sites, I can infer which of the next-hops was selected based on routing policies.

In order to identify the "decision point" AS, I need to infer AS-level paths from traceroutes collected at RIPE Atlas probes. CAIDA's prefix-to-AS mapping datasets [93] provide basic mapping from IP-level traceroutes to AS-level paths. But this simple mapping method is inaccurate and incomplete because of missing hops, multiple-origin prefixes, and the IXP prefixes in traceroutes. I then describe how I infer AS-level path from traceroute path.

Mao et al. [98] proposed a heuristic method to improve IP-to-AS mapping. They collected traceroute and BGP tables from the same set of vantage points. Then, they proposed algorithms to identify various factors that may cause missing and extra AS hops observed in traceroute by comparing the traceroutes and BGP AS paths. Without BGP feeds from RIPE Atlas probes used for traceroute measurements, their algorithms do not apply directly. However I can apply their methods to refine AS path inference:

- If an unresponsive/unresolved IP hop from traceroutes is between of two hops that map to the same AS, I assume the unmapped hop belongs to the same AS as the surrounding AS hops.

- If an unresolved IP hop is in between hops that map to different ASes, use the domain name of the unresolved IP hop, if available, to associate it with a

neighboring AS.

- Identify prefixes that belong to IXPs. IP addresses assigned to IXPs may appear in traceroutes and thus introduce an extra AS hop relative to the corresponding BGP AS paths. I identify such hops and remove them from inferred AS path. Nomikos and Dimitropoulos provide a tool [99] to collect IP prefixes assigned to IXPs. They collect data from PeeringDB [100] and PCH [101], including prefixes for over 1000 IXPs. Using this dataset should yield better detection accuracy than the algorithm for IXP detection used in [98].

- Detect multiple origin ASes (MOAS). Once found a MOAS hop, I map it to a set of ASes. For the rest of the paper, I include these traceroutes in our comparison with other traceroutes. I consider these traceroute hops "match" with the corresponding hop in other traceroutes if the AS in the other path matches any one of the ASes associated with the MOAS hops.

According to the evaluation in [98], with basic IP-to-AS mapping using BGP tables, only about 72% of traceroutes matched the corresponding BGP AS paths. By applying the first three steps above to resolve the unmapped IP hops and IXP addresses, the matching rate increased above 80%. Based on this, I expect that applying these techniques will infer the AS path with at least 80% accuracy. Note that this overall matching rate serves as a lower bound on the matching accuracy of suffixes of the paths (after the "decision point").

I do not consider traceroutes that cannot be completely resolved: if an un-

responsive or unresolved IP hop lies between two different ASes, I abandon this traceroute and the comparison to other paths from the same probe. This affects at least one traceroute from 20% of the probes for C and D root and from nearly half of the probes measuring K, described in more detail below in Section 4.4.

4.4 Measurement methodology

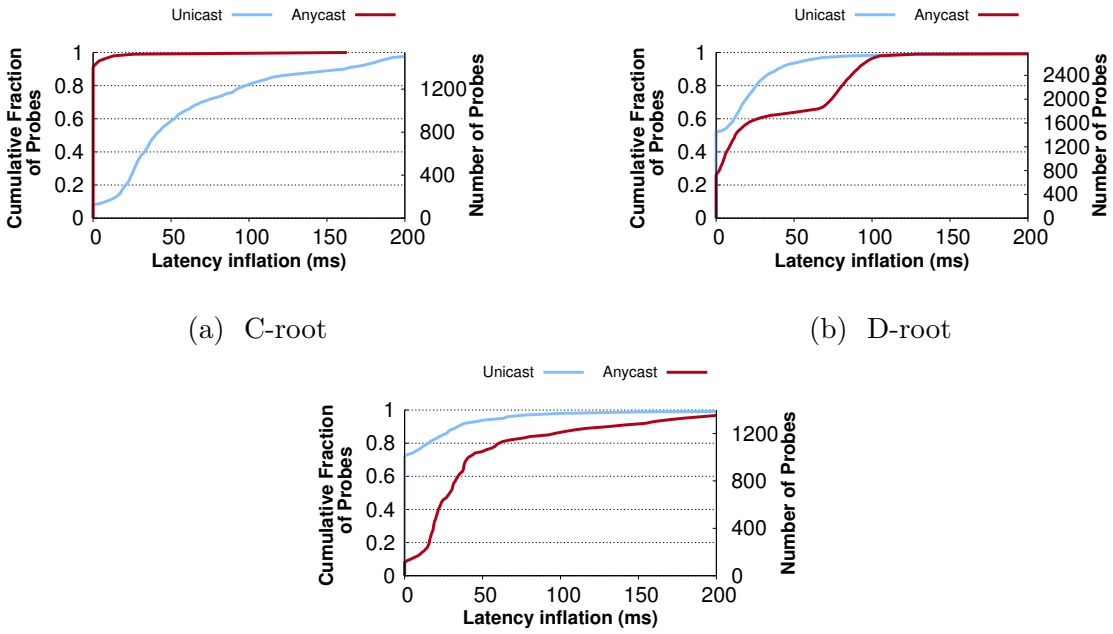
In this section, I describe the experiments conducted to help quantify anycast path inflation. Suppose source s sends a query to anycast address a; this query reaches site $S_{s\to a}$. With methods described in §4.2 and §4.3, I can evaluate the performance of an alternate site that *could* have been chosen for the query.

In this experiment, I collect performance measures from each RIPE Atlas probe s to three particular anycast sites: its selected anycast site $S_{s\to a}$; its geographically closest site $G_{s\to a}$; and the site that provides s lowest latency $L_{s\to a}$. $S_{s\to a}$ is already measured by RIPE Atlas probes with the "built-in" traceroutes. $G_{s\to a}$ is easy to measure by sending traceroute to the unicast representative of the nearest site to the RIPE Atlas probe. Ideally, $L_{s\to a}$ should be obtained from exhaustively probing each anycast site from the probe s. However, RIPE Atlas platform is a shared resource that enforces rate limiting. Due to limited measurement budgets on RIPE Atlas platform, it is not feasible to conduct exhaustive probing. Instead, I measure a couple more candidate anycast sites, and set $L_{s\to a}$ as the one that provides lowest latency among measured sites. Note that it only causes (potentially) underestimation of anycast path inflation by sampling candidates for $L_{s\to a}$ As shown in later

results, anycast path inflation is already high without exhaustively maximizing the inflation.

Specifically, I estimate the lowest latency from s to a as follows. If the measured latency to the geographically closest site $G_{s\to a}$, is less than that predicted by distance (using the Htrae constant [90], 0.0269 ms/mile) to the second closest site $G'_{s\to a}$, then assume $L_{s\to a} = G_{s\to a}$. That is, choose the geographically closest site as the lowest-latency site if the second closest (so are other sites) is unlikely to be better. If $S_{s\to a}$ is already the second closest replica $G'_{s\to a}$, assume $L_{s\to a}$ is either $S_{s\to a}$ or $G_{s\to a}$, whichever is less. Otherwise, I will measure the latency to $G'_{s\to a}$ and set $L_{s\to a}$ to the least-latency one of measured sites $S_{s\to a}$, $G_{s\to a}$, and $G'_{s\to a}$. In some cases, I may choose to measure the third-closest, or a popular site that is within a distance that could yield lower latency.

I focus on the probes whose selected anycast site $S_{s\to a}$ is farther than 500 km beyond the geographically closest site $G_{s\to a}$. For these probes, it is likely that their performance can be improved. For C-root, I collected traceroutes from 1862 such probes, and 1541 of them have all complete traceroutes according to §4.3; for D-root, I collected traceroutes from 3570 probes and 2785 provided complete traceroutes; for K-root, I collected traceroutes from 2886 probes and 1398 of them were complete. With these measurements, I quantify anycast path inflation and compare it with unicast path inflation in the next section.



(a) C-root

(b) D-root

(c) K-root

Figure 4.3: Comparison between anycast path inflation and unicast path inflation.

4.5 Quantify anycast path inflation

With the traceroutes measurements to $S_{s\to a}$, $G_{s\to a}$ and $L_{s\to a}$ for each probe s, I analyze how much of the performance deficit in anycast is due to unicast path inflation, and how much is due to anycast path inflation caused by bad route selection.

Anycast path inflation is computed as the difference between latencies to $S_{s\to a}$ and $L_{s\to a}$. Typical, unicast path inflation from BGP is computed as the difference between the latency to $L_{s\to a}$ and the predicted latency with Htrae constant [90], by distance, to $G_{s\to a}$.

Figure 4.3 shows unicast and anycast path inflation measured from 1541 probes for C-, 2785 for D-, and 1398 for K-root. The results show that for C-root, anycast

path inflation is much smaller than unicast path inflation. As described in §3.4, once the queries to C-root entered Cogent, anycast provider of C-root, it is unlikely that queries are directed to distant sites based on intra-domain routing. For D- and K-root, anycast path inflation is larger and affecting more probes than unicast path inflation. That is, for D- and K-root, BGP routing often fails to choose the better route for a probe, and incurs larger latency inflation by send queries to a distant site. Consider that D- and K-root have more anycast sites, it seems suggest that extra choices provided by more sites can *harm* performance, since ISPs may (and do) choose the worse route out of many available, thereby increasing the latency to the anycast.

I further breakdown the anycast path inflation by it specific causes. As described in §4.1, anycast path inflation is incurred by selecting a worse route to a distant anycast site among the available ones. With the traceroutes from each probe to multiple anycast sites, I infer the AS paths selected by BGP and the "decision point" where the route selection failed using the method in §4.3. Route selection mechanism at the "decision point" is usually based on two policies in the Gao-Rexford model [25, 26] and a common assumption that shorter routes are preferred:

- "Valley-Free": After a provider-to-customer edge or a peer-to-peer edge in the AS path, the route can not traverse customer-to-provider edges or another peer-to-peer edges.
- "Prefer-Customer": The ISP prefers routes through its customer ASes over the peer ASes, over its provider ASes.

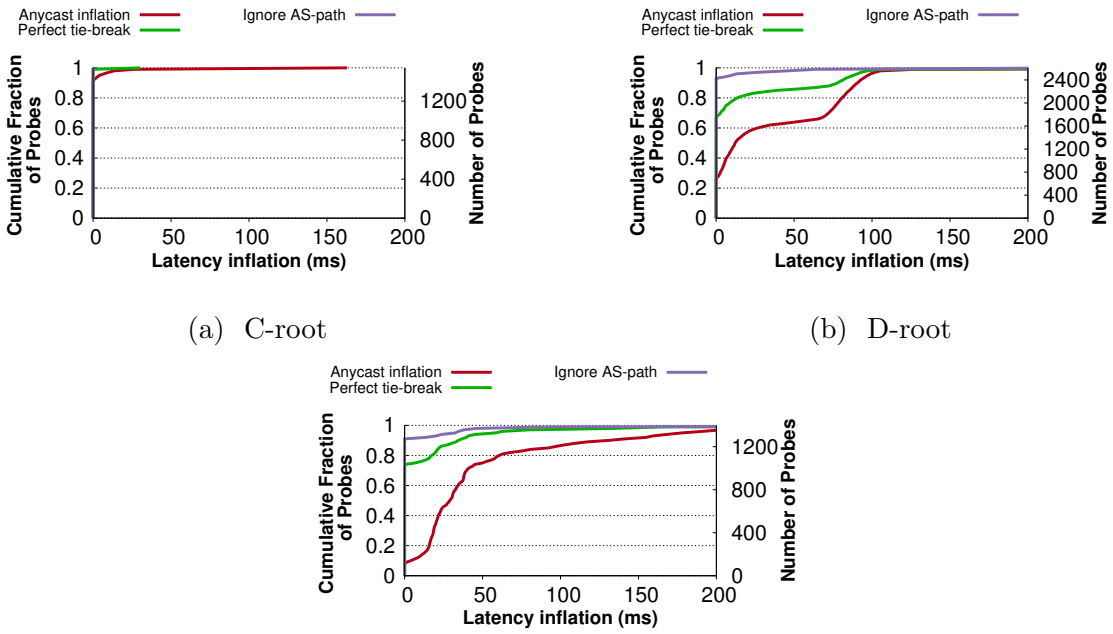
Roots	Total	Good	Prefer Customer	Shortest AS-Path	Unknown Tie-breaking
C-root	1541	91.0%	0.0%	0.2%	8.8%
D-root	2785	26.5%	6.8%	25.5%	41.1%
K-root	1398	8.6%	8.7%	17.3%	65.4%

Table 4.2: Why probes do not choose closest sites.

- "Shorter AS-Path": The ISP prefers routes with shorter AS path length.

Note that the available paths to different anycast sites are policy-complaint paths, i.e., all these paths are "Valley-Free". For each probe, I compare the available paths to different sites at the "decision point", and identify which policy likely caused BGP to make the decision: "Prefer-Customer", "Shorter AS-Path" or some "Unknown" tie-breaking rule if the first two policies result in a tie. First I obtain AS relations between "decision point" AS and its next-hop ASes in different routes with AS-relation datasets [93], and order AS relations based on "Prefer-Customer" policy. If the selected route is preferred, then I claim the route is selected due to "Prefer-Customer". If the selected route is not preferred based on "Prefer-Customer", I compare the AS path lengths of different routes starting from "decision point". If the selected route is shorter, it is probably selected due to "Shorter AS path" policy. Otherwise, I conclude that the route is selected based on some "Unknown" tie-breaking rules.

Table 4.2 shows the results of the breakdown of causes of anycast path inflation.



(a) C-root

(b) D-root

(c) K-root

Figure 4.4: Anycast path inflation and inflation if BGP can: (1) 'tie-break' correctly; (2) also ignore AS-Path length.

It lists the number of measured probes to C-, D- and K-root, the number that were routed to the lowest-latency sites, and the numbers that were not due to various reasons. For all three roots, the results suggest that the queries were not directed to the lowest-latency anycast site mostly because of some "Unknown" tie-break rules. In other words, many of the queries could have been routed to lower-latency sites if the ISPs applied better tie-break rules. I will next quantify the benefits of better tie-break rules and route selections.

Figure 4.4 shows how much of the anycast path inflation can be recovered if "decision points" select routes better. The figure shows results from the same measurements as in Figure 4.3 for C-, D- and K-root: the "Anycast inflation" (red)

lines correspond to the anycast path inflation as shown in Figure 4.3. The "Perfect tie-break" (green) lines illustrate the anycast path inflation that remains if "decision points" always select lowest-latency routes among those who have the shortest AS-path length. The "Ignore AS-path" (purple) lines show anycast path inflation when the "decision points" always select the best route regardless of AS-path length, but still follow "Prefer-Customer" policy.

Table 4.2 and Figure 4.4 are extremely encouraging results: they show that much of the performance deficit can be recovered if "decision points" ISPs tie-break more intelligently. BGP routers make selections mainly based on policies, and do not have sufficient information to make good selections. There exists measurement-based optimization services that help identify lowest-latency route, and such services are used for multi-homed ASes to choose providers (e.g.,Internap Managed Internet Route Optimizer [102]). However, the deployment of such services needs to be widespread enough in order to systematically improve the performance of anycast, which requires long time and large resources to achieve. In the next section, I will discuss what other forms of fixes can be applied on BGP to improve anycast.

4.6 Conclusion

As anycast deployed in more critical infrastructure, understanding the problems in its performance and the root causes becomes an important task to maintain and improve efficiency of anycast. However, prior work provides few explanations on the root causes for anycast's inefficient performance. Researchers expect BGP routing

to be at fault for performance deficit, but rarely quantitatively show how much of the deficit BGP causes.

In this chapter, I describe that how does anycast, in addition to unicast, introduce another layer of path inflation. Although prior work has shown that BGP may create circuitous paths and thus path inflation [92], I develop a novel measurement method to quantify how much path inflation is in anycast. By evaluating performance of alternate anycast sites, I compare the path inflation in anycast to that in unicast, and show that anycast inflation is usually much larger than unicast inflation. Further, I apply the standard BGP model [25, 26] to analyze how ISPs end up selecting the route to distant anycast sites. My analysis shows that most of the anycast inflation is due to ISPs' unknown or random tie-breaking among routes that are "equivalent" based on the BGP model. Finally, I show it is promising to fix anycast: much of the performance deficit in anycast can be recovered if ISPs could have information to tie-break more intelligently. In the next chapter, I investigate the causes, if any, that make ISPs tie-break poorly.

Chapter 5: Inferring Provider Preferences via Route Manipulation

In the previous chapter, I conclude that the large latency inflation observed in many anycast deployments is mainly caused by performance-unawareness of BGP routing. Gao and Rexford [25, 26] proposed the dominant model of BGP routing. They model interdomain routing with two rules: "Valley-free" and "Prefer-Customer", as described in section 4.5. When multiple routes satisfy the two rules, researchers usually assume that the routes with shortest AS-path length are selected, i.e., the "Shorter AS-path" policy. But is this assumption always true? Even if the shortest paths are preferred by ISPs, since the paths are typically short and may be shorter among busy paths due to the prevalent usage of CDNs [103, 104], one should expect *tied* routes exist, especially in anycast scenarios [68]. Prior work on modeling Internet routing has used different ways to break ties: by AS number [83], by customer cone size [105] or at random [64, 74].

To better understand the problems I discovered in previous chapters, i.e., ISPs fail to choose routes to the nearby anycast sites among the equivalent ones, I develop methods that use large-scale data plane measurements and control plane experiments to identify how ISPs choose providers to forward their traffic when presented with valid routes from multiple providers. The insight is that by selectively

making certain paths look better or worse to a particular ISP, I can infer the ISP's preferences relative to alternative paths. I conduct experiments on the PEERING testbed [27, 106] using several route manipulation techniques:

Route poisoning (§5.2) My first experiment exposes alternate paths to PEERING prefix from different ASes, and reveals relative preferences among the paths. I use PEERING to temporarily poison BGP announcements and cause ASes to withdraw the poisoned routes, and then un-poison the routes. I run traceroutes from the vantage points towards the PEERING prefix to observe the paths before and after I poison/un-poison the routes. This experiment allows us to discover relative preference among paths and thus reverse engineer route selection process.

Community tags (§5.3) In the second experiment, I use PEERING to embed in BGP announcements a set of community tags customized for traffic engineering in different ASes. The community tags, if propagated to the target ASes, will cause the ASes to not export the routes, thus effectively "poisoned" routes. Same as in the first experiment, I then obtain alternate paths and reveal relative preferences.

Selective prepending (§5.4) In my third experiment, I selectively un-prepend and prepend AS paths from PEERING, and then send traceroutes from vantage points to the prefix. I discover the ASes that prefer shorter AS-length routes with measurements from this experiment.

For all of the experiments, I use the RIPE Atlas platform [87,88], which consists a set of approximately 9,000 probes located in about 180 countries and over 3,500 ASes, as the vantage points to collect traceroutes.

Once I discover relative preferences among routes, I focus on ones that are tied in Gao-Rexford model, and the preferred routes are *not* due to "Shortest AS path" policy. I thus reveal that customer ISPs' relative preferences among their providers. Such preference rankings among Tier-1s are configured through 'Local Pref' in BGP, and are considered before the "Shortest AS path" policy in route selection. By aggregating the discovered preference ranking among Tier-1s, I find common and consistent partial order of Tier-1 ISPs for their customers: if one customer ISP prefers Tier-1 x over Tier-1 y, many other networks do, too. I further reveal the customer ISPs who select the shorter routes from multiple Tier-1s. However, only about 50% of the customers choose to use shorter routes when available, much of other customers remain on the longer paths. These results suggest the "Shortest AS path" policy might be applied less often than people expected, thus interdomain traffic engineering that employs selective AS path prepending to redirect traffic may not be as effective [11]. In the end, I quantify the effect of the discovered preference to anycast performance.

5.1 Motivation and background

5.1.1 Funneling effect of transit providers

Observations in Chapter 3 on query distributions across Tier-1 ISPs motivate us to consider further investigations in how customer ISPs choose among routes received from multiple Tier-1s. For example, even though the closer to European anycast sites exist, many clients choose to send their packets particularly through Deutsche

Telekom and then directed to the D-root site in Virginia. It seems that for those clients, Deutsche Telekom is the preferred Tier-1 to send packets through. Are there local policies installed within customers of Deutsche Telekom that encourage them to choose the routes through it, even those routes might perform poorly? In anycast, if customer ISPs tend to prefer one provider who only have routes to poor anycast sites, it may significantly harm anycast performance. As shown in Figure 3.11, a Tier-1 provider (e.g.,DTAG, KPN, etc.) might always "funnel" queries through it to a certain D-root site *mcva*. And it happens not only for D-root.

I use following measurements to demonstrate such "funneling" behavior is common for other DNS roots in a research/educational transit provider, Internet2 [107]. Internet2, along with some other research and educational networks (e.g., GEANT(AS20965,AS21320) [108]), provide cheap or even free transit for university/research networks [109, 110]. As ISPs tend to send their traffic through a cheaper neighbor [59, 60], it is likely that ISPs prefer to use routes through Internet2. As a preferred transit provider, how Internet2 route queries to anycast prefixes would have large impact on its customers. I collected RIPE Atlas traceroutes to nine DNS roots that went through Internet2 , and tracked where the queries are routed to. I used the traceroutes from two separate days, May 1st 2016 and May 1st 2018, to exclude transient observations. As shown in Table 5.1, Internet2 funneled the queries through it to one particular anycast site for most of the roots: On May 1st 2018, traceroutes to 6 out of 7 DNS roots through Internet2 are all directed to one anycast site of each root; on May 1st 2016, traceroutes to 4 out of 5 roots are funneled to one site of each root. This result suggests that Internet2 funnels their

queries to particular anycast site regardless the sources. Anycast deployments could suffer from the funneling effect caused by transit providers. The more popular these providers are, the more performance deficits in anycast are introduced.

To understand if ISPs have preference towards particular transit providers (e.g., DTAG in Figure 3.11), I work on revealing relative preferences ranking among Tier-1 ISPs for customer ISPs. In the rest of the chapter, I describe how to use route manipulation techniques as measurement methods to answer the following questions: do ISPs have preferences ranking among their providers? Are the preferences ranking among Tier-1s common and consistent across customers? With the relative preferences in place, how often is "Shortest AS path" policy applied in route selection? how do such preferences affect performance in anycast?

5.1.2 The PEERING testbed

My experiments are made possible by the PEERING testbed [27, 106]. PEERING is an experimental platform that allow researchers to interact with the Internet's control plane. It owns a public IP address space and an ASN that researchers can announce to the Internet. PEERING now has 10 servers distributed in Europe, North America and South America, and its prefixes can be announced from all the servers simultaneously. That is, the testbed itself can be used to setup an anycast deployment. These servers peer with hundreds of networks that provide rich connectivity to the clients and vantage points. PEERING provides functionality such as announcing poisoned route (i.e., inserting poisoned AS in the announcements), em-

Roots	May 1st 2018			May 1st 2016		
	Popular site	# Probes to popular site	# Probes traversed I2	Popular site	# Probes to popular site	# Probes traversed I2
A-root	*lax*	5	5	none	0	0
C-root	none	0	0	none	0	0
D-root	*nyny*	1	1	*mcva*	78	78
E-root	*nuq*	49	49	*arc.nasa*	38	38
F-root	none	0	0	none	0	0
I-root	*chi*	50	51	*chi*	45	45
J-root	*yvr*	22	23	none	0	0
K-root	*ch-gva*	53	53	*ch-gva*	33	49
L-root	*lwc*	39	39	*syd*	25	25

Table 5.1: Which site did Internet2 route the queries to?

bedding community tags in the announcements, and announcing prepended routes at selective locations.

I use this testbed to announce prefixes to the Internet from all 10 locations in the experiments in an ethical manner. While conducting I strictly follow the PEERING Acceptable Use Policy (AUP) [111]: I only announce prefixes that belong to PEERING and are allocated to us (no other users were using our prefix). Moreover, I always announce prefixes with the correct origin ASN (i.e., the one it owns) to make sure no hijacking took place in the experiments. I make BGP announcements from the testbed at the rate of at most once per 20 minutes to allow route to converge

and avoid route flap dampening. In short, I use PEERING as instructed; I make sure my experiments do not overload BGP routers in the Internet with excessive BGP announcements; and I do not adversely affect any other networks' routes.

5.2 Route poisoning experiment

My first experiment uses route poisoning to selectively make alternate routes preferable. By strategically announcing poisoning routes, this experiment reveals many ASes' relative preferences among routes and amongst their providers amongst their providers. The experiment proceeds in three phases:

- **Default** I initialize the experiment by advertising a /24 prefix S (allocated to us by PEERING) from 10 PEERING locations. S does not host any services, and thus, advertising, withdrawing, or poisoning route to P does not affect any material traffic on the Internet. After advertising the prefix, I collect traceroutes from RIPE Atlas probes to an address in S. These traceroutes, mapped to AS paths (using AS path inference as described in 4.3 to map the traceroutes to AS paths), gives me the default paths chosen by different ASes.

- **Poison** I take the top-50 ASes that are traversed the most in the default traceroute set, and advertise "poisoned" paths, such that these ASes cannot forward traffic to P. Specifically, one by one, for each AS a in the top 50, I advertise routes in which we insert a in the route advertised by PEERING; this eventually triggers loop prevention in BGP and causes a to dislodge the default route, as a must not forward the "poisoned" route since it's AS number

is already in the route received. After each poisoned advertisement is sent, I again collect traceroutes (and map them to ASes) to discover alternate AS paths (if any) to P.

- **Recovery** Finally, I restore the poisoned routes by re-announcing the original BGP announcements (without poisoning). Once again, I conduct traceroutes from the same set of RIPE Atlas probes and map them to AS paths. This final set of AS paths lets us determine if the alternate paths remain preferable after the original (default) paths becomes available again.

Similar route poisoning experiments have been used as measurement methods to discover hidden network topology [80], to evaluate the prevalence of default routing [81] and to identify alternate back-up paths in the Internet [79]. Researchers have also proposed to use BGP poisoning as a mechanism to reroute traffic and avoid congestion links under DDoS attacks [82–84]. I use the route poisoning's ability to reveal local preferences of ASes, and augment it with other route manipulation mechanisms, including remote traffic engineering with community tags and selective AS-path (un)prepending.

While running the route poison experiments to discover alternate routes, I collect the ICMP packets from RIPE Atlas probes on a PEERING client. The client runs a BIRD routing daemon and connects to each PEERING server via an OpenVPN tunnel created between a local TAP interface and the server. PEERING prefix is announced from the client via BIRD routing daemon to the servers, then to the Internet. Traceroutes towards the prefix will be routed to the PEERING

servers, and then to the client. With this deployment, I can collect the packets that are sent to the prefix locally on the client I run. By collecting traceroute packets on the host (using tcpdump), I identify which server receives each packet by checking which TAP interface received the packet. The collected packets allow me to identify which servers the default and alternate paths are lead to, and thus evaluate the anycast performance of the default and alternate paths.

5.2.1 Inferring provider preference

For each poisoned AS and for each RIPE Atlas probe used, I obtain a path triplet: a default path, an alternate path and a recovery path. In some cases, the alternate path is null, since the probe only had one path to P or all other paths to P become unavailable after poisoning. Using these, I reveal the preferences among the triplet of paths, and infer whether ISPs prefer routes from one provider over others, regardless of AS path length. Relative preference between paths is established if 1. the recovery path is the same as the default path, and 2. the alternate path has shorter AS path lengths. Indeed, it is possible that local policy at an AS could still explicitly prefer paths from a provider when one or both of the previous criteria is violated; this experiment is unable to positively assert such preferences. From the preferences between default and alternate paths, I infer the "decision point" ISP's preference among its providers. I focus on Tier-2 ISPs as they reveal preferences among Tier-1 providers.

I conducted the three phase route poisoning experiment monthly over five

Date	# probes	# Src. ASes	# Path Triplets	Not Reverting to Default	Reverting to Default	Default not Shorter
Dec. '18	3178	3063	982	28	964	349
Jan. '19	3202	3073	1059	17	1042	381
Feb. '19	8325	3080	2203	48	2155	648
Mar. '19	8489	3099	1995	31	1964	553
Apr. '19	8459	3107	2154	64	2090	615

Table 5.2: Route poisoning deployments

Date	No Alternate	Not Reverting to Default	Not Longer Alternate	Longer Alternate
Dec. '18	169	15	76	41
Jan. '19	139	9	80	40
Feb. '19	179	25	64	49
Mar. '19	205	8	44	49
Apr. '19	204	28	70	20
All	232	52	136	74

Table 5.3: Breakdown of Tier-2 ISPs' policies on selecting providers, as recorded by our monthly experiments.

months, once each in December 2018 through April 2019. I collect the path triplets and conduct analysis described in the last paragraph. Table 5.2 shows the details of the deployments as they changed over time: the "# Probes" and "# Src ASes"

columns list the number of RIPE Atlas probes used in my experiments (and their source ASes). The "# Path Triplets" column counts the number of complete (Default, Alternate, Recovery) path triplets these experiments generated. I exclude paths triplets that cannot be fully resolved to AS paths, and triplets whose default paths do not traverse the poisoned AS or alternate path still traverse the poisoned AS (likely due to default routing [81]). The "Reverting to Default" column counts the number of configurations in which the probes reverted back to the default path when it became available again. This result shows that for over 90% of the path triplets, the default paths are preferred over the alternate. The "Not Reverting to Default" columns counts the configurations in which the default path was not reverted back to. The latter counts cases in which the AS kept the alternate path or chose an entirely new recovery path. Finally, the "Default not Shorter" column counts the cases in which the default AS path was not shorter than the alternate paths. There are about 30% of the complete triplets reveal the preferences towards default paths even the alternate have shorter or equal AS path lengths.

Further analysis of the "Reverting to Default" routes enable us to understand whether ASes prefer certain providers, regardless of the advertised AS path lengths. I focus on decisions at multi-homed Tier-2 ISPs who use multiple Tier-1 as transit providers. Tier-1 providers have global presence and many hundreds of peerings and customers. Thus a large number of routes, especially routes connecting two distant hosts, traverse Tier-1s. Preference among Tier-1s, when exists, will likely be revealed by customers ASes in my experiments. In Gao-Rexford model, Tier-1 ISPs are considered equivalent since most other ISPs treat them as providers.

However, they might have vastly different performance, as shown in Section 3.7. The preference between Tier-1s provides important insights on modeling routes selection process and understanding the impact on routing performance.

I catalog whether the multi-homed Tier-2s reverted to the default path *even if* the alternate AS path was shorter. Table 5.3 breaks down the behavior of Tier-2 ASes, each of which have more than one Tier-1 provider. The "No Alternate" column counts Tier-2 ASes that have only one path to P, regardless of being multi-homed according to CAIDA's AS topology data [93]. These ASes lose their route to P during the "Poison" phase of the experiment. The "Not Reverting to Default" column counts the Tier-2 ASes that do not revert to their default path after it becomes available again. The "Longer Alternate" counts Tier-2s that find alternate paths, but these paths are longer than the default. For the ASes in these two columns, I cannot infer Tier-1 preference from this experiment.

The "Alternate not Longer" column counts Tier-2 ASes that revert back to the default path, even if the alternate path was the same length or longer. This column shows that these Tier-2 ASes prefer paths through the default Tier-1 provider over comparable or shorter paths from another. In the following subsection, I focus on these Tier-2 ISPs and examine if their preferences among Tier-1 providers are common and consistent.

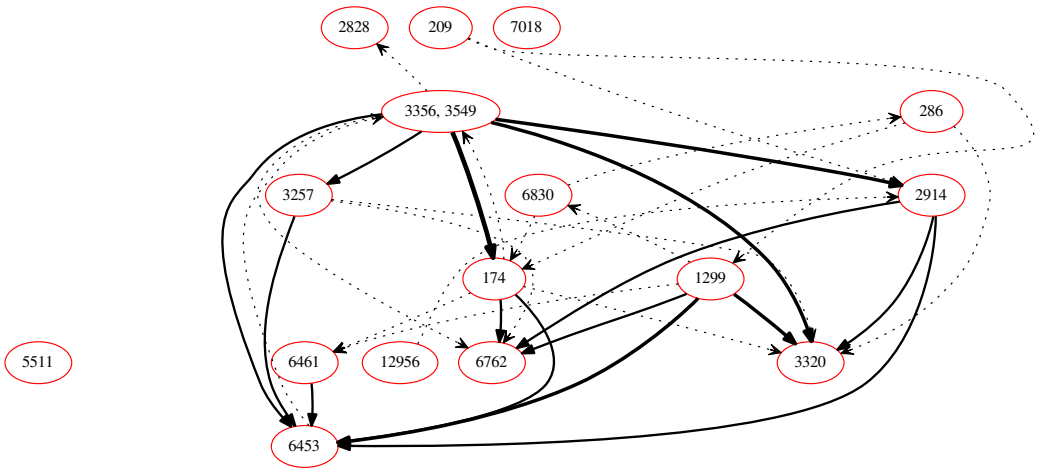
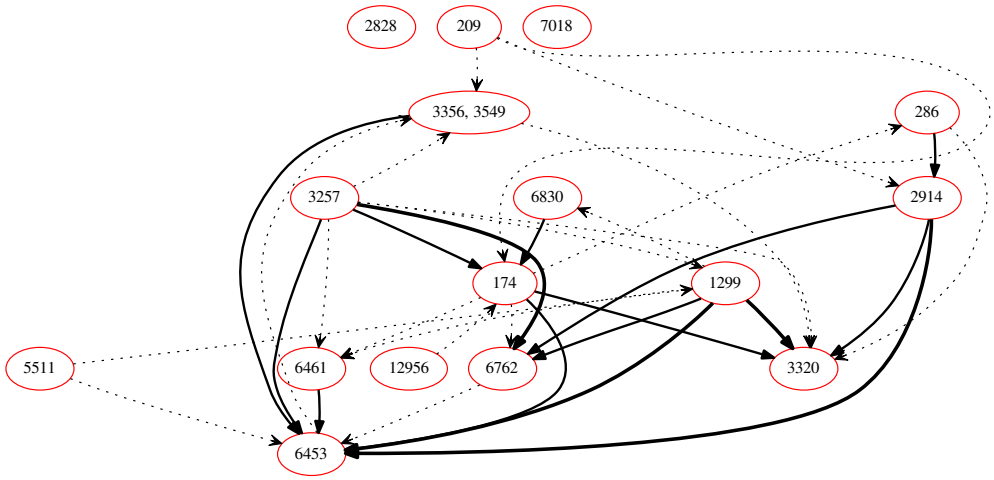


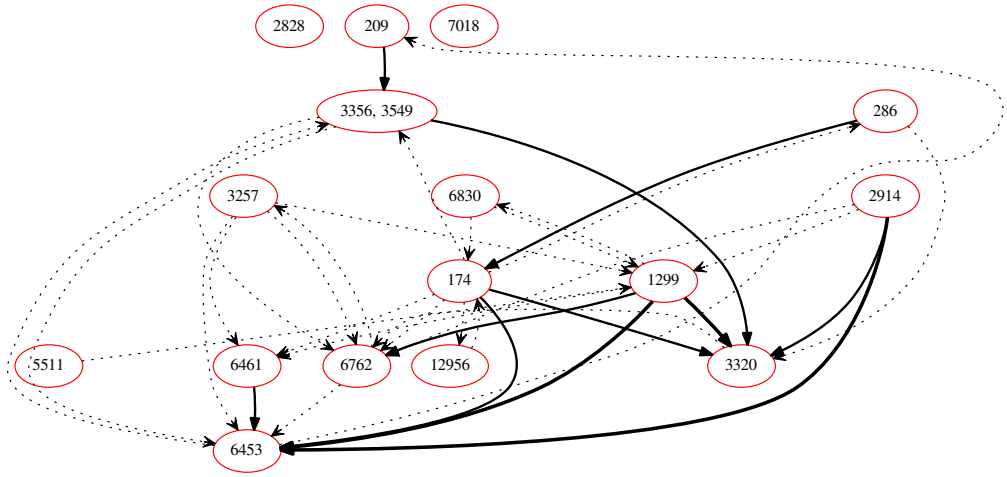
Figure 5.1: Preferences between Tier-1 ISPs revealed by the experiment in December 2018. In the graphs, nodes represent ISPs and edges represent observed preferences between the ISPs. Edge direction shows preference order (from high to low), edge width indicates number of ISPs that have the preference, and dotted edges represent preference had in only one ISP.

5.2.2 Preference ranking among Tier-1 ISPs

The "Alt. not longer" column in Table 5.3 counts Tier-2 ASes that prefer paths though Tier-1 providers, regardless of AS path length. Such preference is not a result of the "Prefer Customer" policy since the Tier-2 ASes in this case are customers of both the upstream Tier-1s. Figure 5.1, 5.1 and 5.3 depict these preferences using a graph: each node is a Tier-1 ISP, and there is an directed edge between two nodes if the source was preferred even though AS path length over the destination in any of the route poisoning experiments. For example, an edge from *AS3356* to *AS3320* indicates that *AS3356* is preferred over *AS3320* by some customer ASes *D*s. The



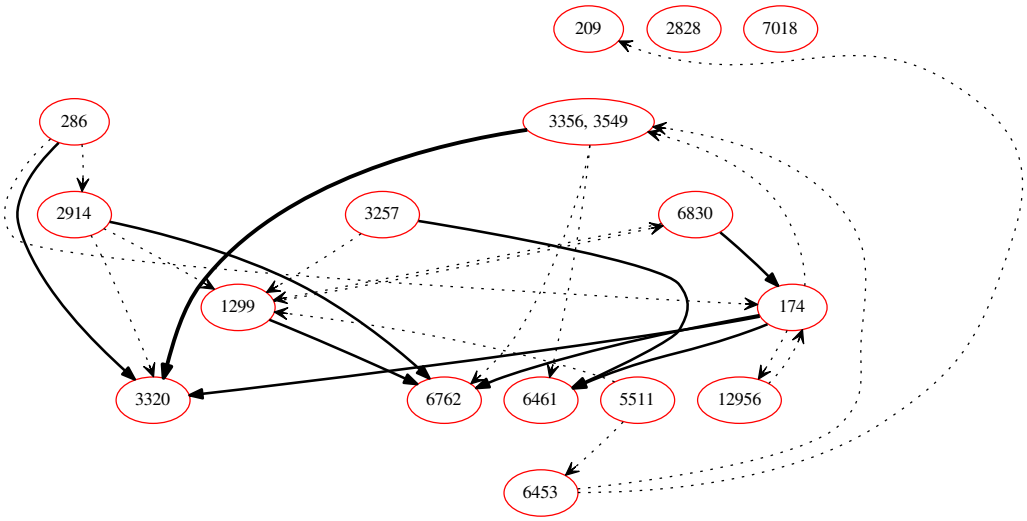
(a) From experiment in January 2019.



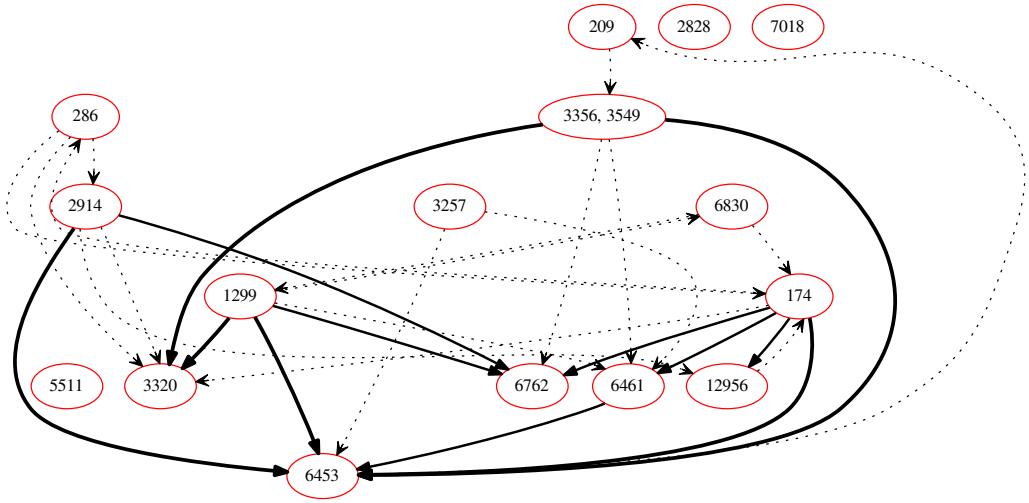
(b) From experiment in February 2019.

Figure 5.2: Preference between Tier-1 ISPs revealed by the experiments in January and February 2019.

thickness of the edge indicates (in log scale) the number of distinct Tier-2 ASes that preferred the source node over the sink. The dotted edges represent preference appeared in only one customer ISP. Figure 5.4 show preferences among Tier-1 ISPs



(a) From experiment in March 2019.



(b) From experiment in April 2019.

Figure 5.3: Preference between Tier-1 ISPs revealed by the experiments in March and April 2019.

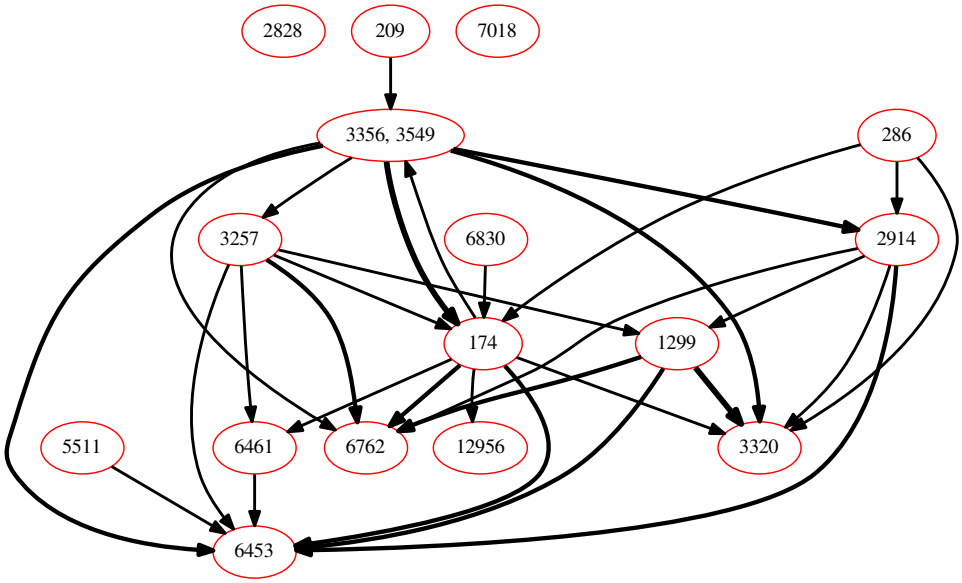


Figure 5.4: Preference between Tier-1 ISPs aggregated over all experiments. Preference revealed through only one ISP is not shown.

aggregated across all experiments. This figure excludes dotted lines, i.e., the cases where only one Tier-2 AS had a preference. If the preference is observed from only one ISP across all experiments, it is likely a local decision specific to that ISP.

The graph shows an unexpected property: from the route poisoning experiments, preferences over Tier-1 ASes form a near perfect partial order, and this ordering is consistent over the five months of data. Only one pair of nodes (AS3356 and AS174) have edges in both directions, indicating that some Tier-2 ASes preferred one over the other, and vice versa. In *all* other cases, either there was no data, or the preferences are consistent across all the Tier-2 ASes I am able to measure. This insight is significant because 1) it implies finer-granularity relationships between ISPs than Gao-Rexford model; 2) as the preferences are configured through

Experiment date	Revealed preference	Default routes lead to sites that			
		Farther	Same	Closer	Unknown
Dec. 26, 2018	349	61	219	64	5
Feb. 7, 2019	638	110	352	137	39
Mar. 10, 2019	553	87	291	95	80
Overall	1540	258	862	296	124

Table 5.4: Impact of preference between Tier-1s on anycast.

'Local Pref' and are considered before other rules, the "Shortest AS path" policy might be applied less often than people expected, thus interdomain traffic engineering that employs selective AS path prepending to redirect traffic may be less effective. 3) the preferences do not conform to any previously conjectured rules for tie-breaking when choosing providers.

5.2.3 Do preferences among Tier-1s affect anycast performance?

In the previous section, I show that relative preferences between Tier-1 ISPs exist and are consistent across many customer ISPs. In this section, I evaluate how these preferences between Tier-1s affect the performance of anycast.

Recall that in the route poisoning experiments, the PEERING testbed announces the prefix from servers in ten different locations. That is, the prefix is anycasted. As described in Section 5.2, I collect packets towards the prefix from

local host that is connected to PEERING servers. The local host sets up OpenVPN tunnels between its TAP interfaces and servers. By identifying the TAP interface each packet reached, I determine which server receives the packet. Thus, I can compute the distance between the packet source (i.e., RIPE Atlas probes) and the anycast servers they are directed to. For each of the route triplets collected in the experiments, it contains routes from the same probe to PEERING servers in potentially different locations. I examine if packets travel longer distance over their default routes than over their alternate routes. If so, the relative preferences towards default routes actually introduce path inflation.

Table 5.4 shows the number of routes triplets with preferred default routes, and how many of the default routes lead to farther, the same or closer sites. It summarizes the results from experiments in December 2018, February and March 2019. (For the other two experiments, in January and April 2019, the process collecting packets on local host was interrupted and generated an incomplete set of mappings between probes and their destinations.) I thus exclude those two experiments in the evaluation. The table lists the number of triplets that revealed preferences, which is the same as "Default not shorter" as in Table 5.2. Among the preferred default routes, around 17% of them lead to farther sites compared to their alternatives, while about 19% of them lead to closer ones. This result shows that, fortunately, the relative preference between default and alternate routes cause only a small portion of probes to reach out farther sites, and a little more probes to use closer sites. Although unaware of routing performance, the policies behind the relative preferences do not introduce anycast inflation overall, nor do they reduce

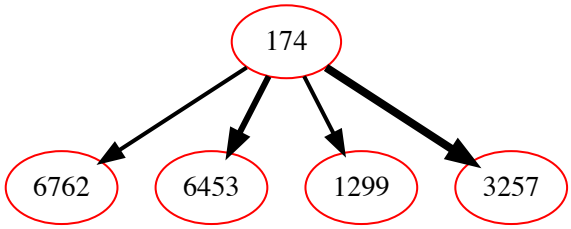


Figure 5.5: Preference between Tier-1 ISPs revealed by 'no-export' community experiments.

the inflation.

5.3 'No-export' community experiment

5.3.1 Poison filtering in Tier-1 ISPs

In the route poisoning experiments, I reveal relative preferences between routes for an AS and thus infer its preferences between its providers. BGP route poisoning enables me to manipulate the routes for its inbound traffic to avoid any particular AS in the upstream, but it has a major limitation: prior studies [83, 84, 112] have shown that some ASes refuse to export BGP routes with poisoned ASes inserted, especially when a Tier-1 is poisoned. In other words, route poisoning one Tier-1 ISP might cause multiple Tier-1s to dislodge their paths to the destination when they receive the poisoned paths. The filtering of poisoned paths will not affect the correctness of the observations of preferences for default paths, but it may damage the completeness of the results because poisoning might fail to discover possible alternate routes through Tier-1s that filter poisoned announcements. I conduct the

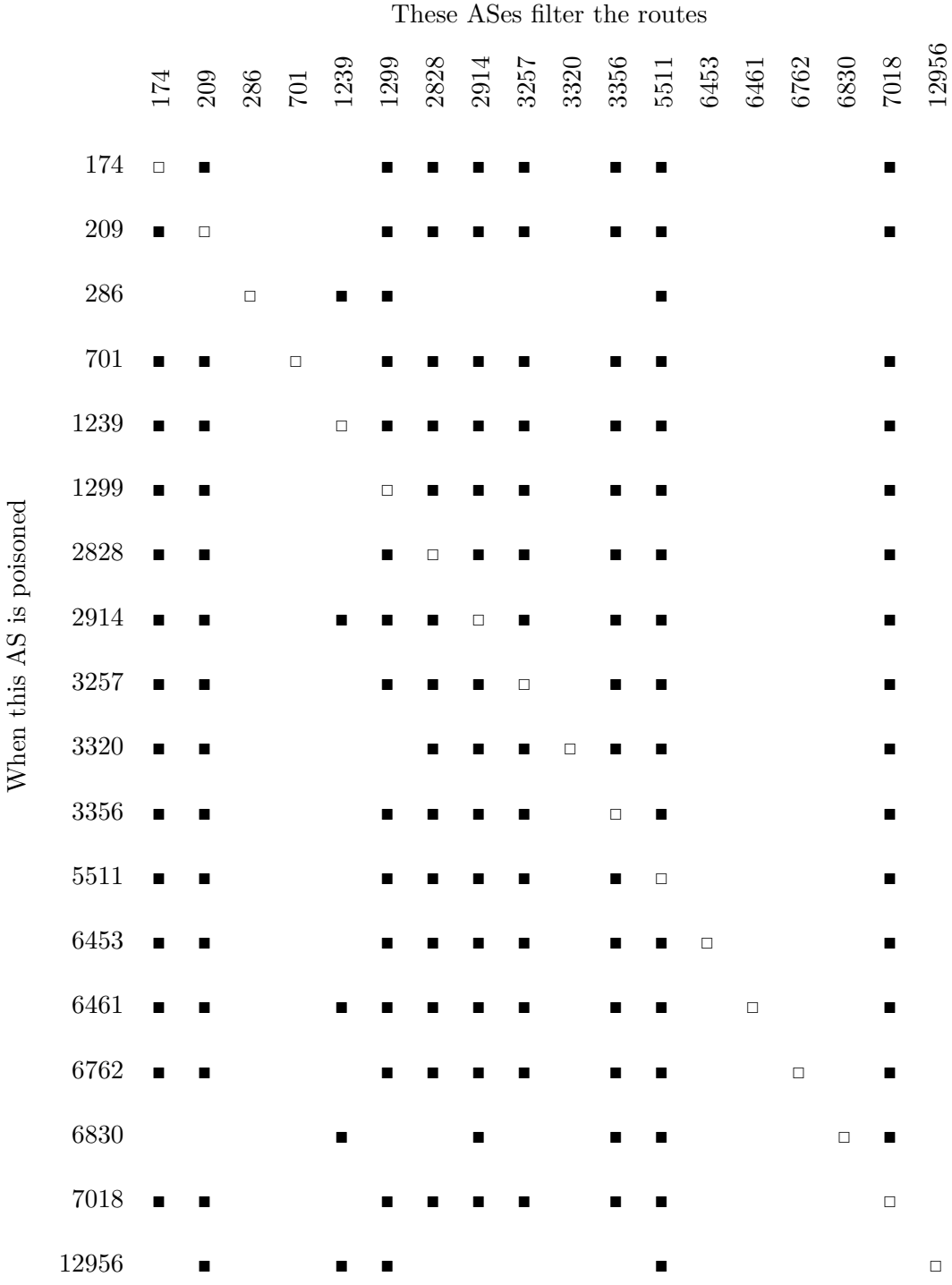


Table 5.5: Routes that falsely include Tier-1 ASes, including poisoned routes, may be filtered by other ASes to prevent misconfigured routes from being used.

following experiments to demonstrate the poison filtering effect and estimate the its impact on previous results. I perform traceroutes before and after I poison a target Tier-1 ISP T. If the number of traceroutes that traversed another Tier-1 F dramatically decreased (by 95% in the experiments), I conclude that the Tier-1 F filters announcements with T poisoned. I use each of the Tier-1s as target ISP to poison.

Table 5.5 illustrates which Tier-1s filter routes include which other Tier-1s. In general, there is substantial collateral damage to poisoning a Tier-1 directly; There is asymmetry in this table: for example, 5511 (Opentransit) filters any route that includes any of the Tier-1s; not all Tier-1's filter routes that include AS5511. Many of route poisoning results thus rely on poisoning selected Tier-2's (in the top 50 most popular ASes) to cause Tier-1's to look for (potentially longer) routes from peers.

5.3.2 Experiment design and results

In an attempt to recover the appearance of poisoning a Tier-1 directly, I designed a complementary experiment that would use per-AS 'no-export' community tags to control the routes towards PEERING. The experiments are similar to those in Section 5.2, but instead of poisoning target ASes to avoid the default paths, I embed 'no-export' community tags for target ASes which effectively cause the target ASes to dislodge their paths. These tags comprise the AS number being configured and a code that specifies which peers should not receive the route. I obtained a list of customized 'no-export' community tags that are public [113, 114]. It is atypical

for an AS to publish tags that are expressive enough to deny export to an AS's customers (not just peers) and are transitive; I found that only AS174, 3549, and 7018 published such tags.

As in the route poison experiments, I collect default paths from probes to the PEERING prefix before sending BGP announcements with the 'no-export' community tags, collect alternate paths after announcing the communities, and recovery paths after re-announcing the original BGP announcements (without communities). Although Streibelt et al. [115] found that most community tags propagate through many ASes, my experiments require *all* routes received by a target AS to carry the 'no-export' tags to fully dislodge its routes to PEERING. I thus selectively announce the prefix from only one PEERING server in Amsterdam to one transit provider (Netwerkverening Coloclue, AS8283) that does not filter community tags. Even though, filtering upstream of this AS may still interfere.

I apply the same analysis as in Section 5.2.1 to analyze the path triplets obtained in the community experiments. In the experiments with community tags from different Tier-1 ISPs, only the experiment that embeds 'no-export' community of Cogent successfully causes it to dislodge its routes to PEERING. Figure 5.5 shows the relative preferences revealed by the 'no-export' community of Cogent. Only one edge between Cogent (AS174) and GTT (AS3257) contradicts with edges in Figure 5.4.

I believe that the community tag approach has potential where poison filtering causes route poisoning to fail. Although it is possible to manually account for the diversity of values needed to ask the AS to accept a 'no-export' route, and some

have this support, community tags are not yet propagated widely enough, nor are they typically powerful enough to substitute for poisoning.

5.4 Selective prepending experiment

The route poisoning and "no-export" community experiments reveal alternate paths and relative preferences among Tier-1 ISPs when Tier-2 ASes select paths. The preferences are presumably configured through 'Local Pref' and are considered before the "Shortest AS path" policy. So the "Shortest AS path" rule might be applied less often than people expected, when a unique preferable path is decided by the relative preferences. As a result, interdomain traffic engineering [11] that employs selective AS path prepending to redirect inbound traffic may be less effective.

Moreover, in many cases of the route poisoning and "no-export" community experiments, unfortunately, the originally chosen path (Default) is shorter, and does not conclusively reveal either AS preferences or selection via preference for "Shortest AS path". The prevalence of relative preferences between ASes may be underestimated. In other words, there might be fewer Tier-2 ISPs than shown in Table 5.3 that choose default paths based on the "Shortest AS path". Thus, I conduct the following selective AS-path (un-)prepending experiments to discover how often the "Shortest AS path" policy is applied by Tier-2s when selecting paths through multiple Tier-1 providers.

	Amsterdam	Seattle	GRNet	ISI	NEU	UFMG	Utah	UW
AT&T	4	3	**6**	3	3	4	4	3
COGENT	3	**5**	**5**	**5**	2	4	3	2
DTAG	2	3	**6**	3	3	4	4	3
GTT	3	3	**6**	3	**6**	**6**	**6**	**6**
KPN	2	3	4	3	3	4	3	3
LEVEL3	**5**	**5**	**5**	**5**	2	**5**	**5**	**5**
NTT	3	2	**5**	2	**5**	3	**5**	**5**
SPRINTLINK	**5**	2	**5**	**5**	**5**	**5**	**5**	**5**
SEABONE	3	3	4	3	**6**	**6**	**6**	**6**
QWEST	**6**	**6**	6	**6**	**6**	**6**	3	**6**
TELIANET	2	**9**	6	**9**	**5**	**5**	**5**	**5**
ZAYO	2	3	3	3	3	4	3	3
OPENTRANSIT	4	3	**6**	3	3	4	4	3
TATA	3	3			**6**	**6**	**6**	**6**
LGI	2	3			3			3
UUNET	3			**6**	**6**	**6**	**6**	**6**

Table 5.6: AS path lengths from Tier-1s to PEERING during selective prepending experiment. Each column represents the site that does *not* prepend. Bold numbers indicate routes directed to any of the prepending locations. Missing values indicate that RouteViews (which may not include a direct peering) included no route through the Tier-1.

5.4.1 Experiment design

PEERING allows researchers to repeatedly prepend origin AS in the advertised paths at selective locations. This feature enables us to control the path length received by ISPs (to a certain extent). I utilize this feature of PEERING and design

the two-phase experiment:

- **Prepend**: In this phase, I prepend the origin PEERING ASN 47065 three times in BGP announcements and advertise the prepended route to a single transit provider of PEERING [116] from each of the 10 locations I use. I do not announce routes to other transit providers other than the 10 selected ones. Once again, I collect traceroutes from RIPE Atlas probes towards the announced prefix and sample a set of 1,000 probes who traversed Tier-1 ISPs. Using these probes, I conduct traceroutes to the PEERING prefix, which gives me the paths chosen when routes are prepended.

- **Un-prepend**: Next I selectively advertise non-prepended routes to the transit provider from one PEERING location each time, and still advertise prepended routes from other locations. These BGP announcements provide shorter AS-path length routes to some ASes. I then conduct traceroutes from the same set of Atlas probes and collect *un-prepended* paths.

These path pairs (prepended, un-prepended) allows me to compare whether path prepending makes a difference in how Tier-2 ISPs choose their upstream Tier-1s. Upon un-prepending from each location, I use RouteViews and Looking Glass servers [117, 118] in each Tier-1 ISP to identify the AS paths from Tier-1s to the destination. The results are shown in Table 5.6. In that table, each column represents the PEERING location that does not prepend. Each row shows the Tier-1's AS path length to PEERING AS when different sites are un-prepended. Some ISPs can be compared by longer or shorter paths; however some pairs (e.g., AT&T and

Change to...	shorter	108
	the same	8
	longer	1
	unknown	7
Remain...	but unknown	19
	and no other T1s	96
	but shorter exists	83
	and no shorter	59

Table 5.7: Number of Tier-2s that changed path, that did not, and breakdown on path length changes.

Opentransit) have identical path lengths regardless of prepending location, so the preferences of a hypothetical customer of both would not be resolvable.

5.4.2 How often Tier-2s choose shorter paths?

Table 5.7 categorizes Tier-2 ISPs based on their behavior with selective prepending at different locations. If the Tier-2 changes from its Tier-1 provider to another after I un-prepended from any one location, I then examine if this new Tier-1 has shorter path according to Table 5.6. If the new paths are shorter, it is likely that the Tier-2 employs "Shortest AS path" when selecting paths from its providers. As shown in the "Change to shorter" column in Table 5.7, for those Tier-2s that changed paths after un-prepending, 108 them change to shorter paths. "Change to same" and

"Change to longer" shows the Tier-2s that make routing changes not due to the shortest path policy, since they do not change (back) to a shorter AS path.

Some Tier-2 ISPs remain on the same paths even after un-prepending. This could be because the Tier-2 had no other provider Tier-1 AS, had only longer paths to other Tier-1 providers or applied a built-in higher local preference towards the Tier-1 that it had chosen. I use AS-relationship dataset from CAIDA [93] to identify Tier-1 providers for each Tier-2 ISP, and use Table 5.6 to examine if there exists shorter paths from other Tier-1 providers available to the Tier-2. In table 5.7, "Remain but shorter exists" show that 83 Tier-2s have shorter paths but chose to remain with their default Tier-1 providers regardless of path length. I obtained similar numbers for "Change to shorter" and in "Remain but shorter exists". This observation indicates that many Tier-2 ASes maintain relative preferences between their Tier-1 providers. Local preference is prioritized ahead of path length in BGP route selection process, and my results show that shortest AS path comes into effect in only about 50% of Tier-2s that have the option to choose between Tier-1 providers.

5.5 Conclusion

The Gao-Rexford model [25, 26] provides a partial order of routes. When it cannot determine the favorable route, it is commonly assumed that a route is selected among the equal good ones by the "Shortest AS path" policy. The model and the assumption is used in studies on analyzing network reliability [63], BGP convergence [64], control of network traffic [65–67], and BGP security [69–71], etc. In previous chap-

ters, I report that much of anycast inflation is due to poor route selections made by ISPs among routes that are equally good in Gao-Rexford model.

In this chapter, I re-examine the assumption in tie-breaking routes: when multiple equivalent routes are presented, do ISPs have a preference or do they choose the shortest one? I used PEERING to develop three control plane experiments—based on route poisoning, community tags, and route prepending—that allow me to directly identify relative preferences among routes from all RIPE Atlas probes.

These set of experiments reveal that ISPs usually apply preferences towards their providers before taking shortest paths. Moreover, I show that Tier-2 ISPs have common and consistent preferences among Tier-1 ISPs, even when they have the same provider-to-customer relations to most ISPs. This is a surprising result: I expected these relative preferences among Tier-1s to reveal business relationships, but it seems unlikely that virtually all Tier-2 ISPs would end up having the same preferences.

Another significant insight from the results is that the "Shortest AS path" policy might be applied less often because ISPs are likely to use local preferences to tie-break in routes from their providers. I find only about half of the Tier-2s choose shorter paths from Tier-1 providers. This observation suggests that selective AS path prepending as a traffic engineering method might not be effective for many ASes, and motivates future studies on evaluating the effectiveness of AS path prepending in traffic engineering.

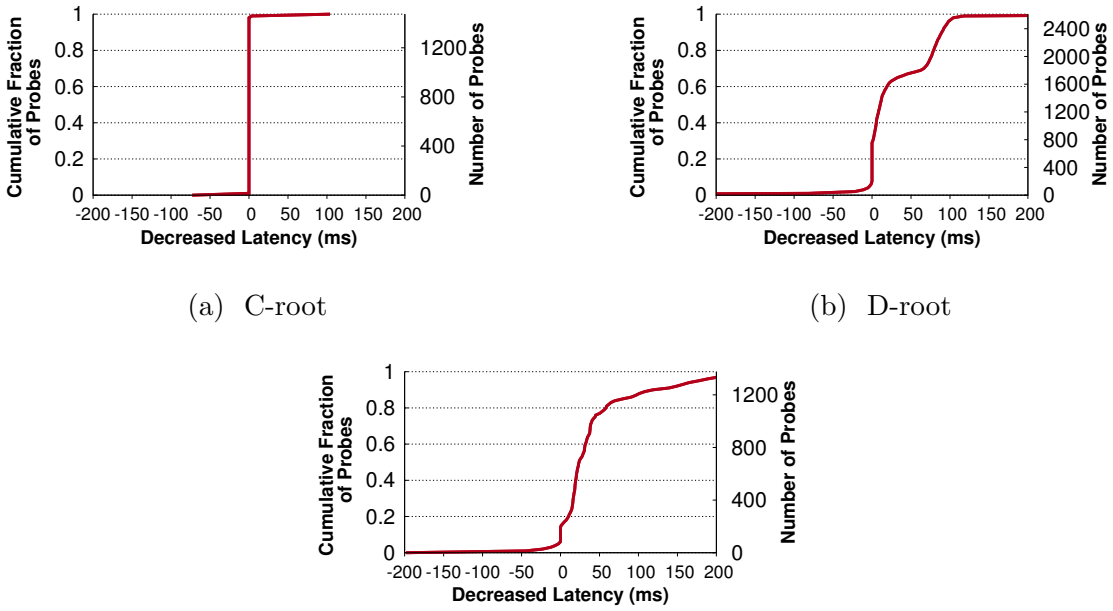
In the end, I evaluate the effect of the newly discovered preferences to anycast performance. Among the preferred default routes, around 17% of them lead to

farther sites compared to their alternatives, while about 19% of them lead to closer ones. Although the policies behind the relative preferences are probably unaware of routing performance, they do not particularly increase (or reduce) anycast inflation.

Chapter 6: Improving Anycast Performance

In this Chapter, I describe an extension to BGP announcement that fixes the inefficiency in anycast performance. With previous results, I show anycast path inflation, which is usually caused by poor route selection, is generally larger than unicast path inflation. Figure 4.4 reports that much of the performance deficit in anycast can be recovered simply by tie-breaking better in route selection process, without violating standard routing policies such as "Prefer-Customer", "Prefer shorter AS-path" and "Valley-Free". However, BGP is policy-based and unaware of routing performance. It lacks information to identify the route that provides better performance in terms of latency or load balance in anycast. For this reason, adding more anycast replicas or sites might not necessarily improve overall anycast performance, but in fact *harm* the performance by making BGP selects the *worse* among more equal routes, as shown in Figure 3.10a and 3.10b.

I will show in the rest of this chapter, that adding static information in BGP announcements for anycast prefixes would be sufficient to recover much of the anycast performance deficit. Fortunately, BGP protocol is extensible and made to support addition information in its announcements. Particularly, BGP community tag [119, 120] is a commonly used field for providing additional information regard-



(a) C-root

(b) D-root

(c) K-root

Figure 6.1: Benefits of geographic hints for different roots.

ing prefixes announced. BGP community is used for traffic engineering [121, 122], mitigate attacks [122, 123], and network troubleshooting [124, 125]. The use of BGP community has become popular by many ISPs in recent years [115]. I will describe how to embed additional information of anycast sites who advertise the prefixes in the BGP announcements, and evaluate how much this additional information can fix the performance problems in anycast. Compared to other suggestions in anycast deployment such as deploying all anycast sites behind a single upstream provider, or measurement-based BGP optimization schemes, this proposed fix is incrementally deployable and introduces little deployment and operation overhead.

6.1 Static BGP hints

In this section, I describe a simple static "hint" that can be embedded in BGP announcements, and show that such a static "hint" can provide large benefit to anycast performance. Consider an extension in BGP announcements for anycast prefixes that includes the geographic locations of sites reachable through the announcements. A BGP router would receive one or more BGP announcements, each advertising one or more sites. When selecting routes, the BGP router may discard some announcements based on standard policies such as "Prefer-Customer". Among the remaining announcements, the router will choose the one that advertises the geographically closest site regarding to the router itself. (If multiple do, then the router may choose one based on some other criteria or randomly.) The router would then re-announce its chosen route to its BGP neighbors, as per usual. All packets through the BGP router to the anycast prefix would be routed through this chosen route to the geographic closest site.

I evaluate this geographic hint through simulation with real traceroute measurements from experiments in §4.4. For each probe in the measurements, I identify the "decision point" AS where the selection among routes to different sites happened. I then consider which sites would be listed in the BGP announcements for different routes, and simulate the route selection process to pick the geographically closest site to the decision point (not necessarily the closest to the probe).

My goal is to identify sites in geographic hints seen from the decision point and choose the closest. I use "undns" [126] to track what locations the traceroutes

traverse, and infer what geographic hints are propagated to the decision point. Consider the example in Figure 4.1, I found the probes' traceroutes to *laca* and *tojp* diverge at Los Angeles. According to the geo-hint, the route selected at the decision point should be the one that leads to *laca*. I compute the latency difference between the geo-hinted site and the chosen site, and characterize the benefits introduced by this hint. Note that for some probes, I cannot obtain the latency to geo-hinted sites. Suppose in the example shown in Figure 4.1, the decision point is located at an anycast site *hkcn*, so the geo-hinted site is clearly *hkcn*. However, there is no traceroute sent to *hkcn* in the measurements I collected, so the performance difference between *hkcn* and the chosen site is not measured. For such probes, I exclude them in the results. For C-root, 67 among 1541 probes have their geo-hinted sites not measured; for D-root, there are 175 such probes out of 2785; for K-root, there are 22 such probes out of 1398.

Figure 6.1 shows the latency improvements for queries to C-, D-, and K-root would obtain using the static geographic list. x-axis is the latency difference; y-axis is the portion or number of measured probes. Latency to D- and K-root both show dramatic improvement. For D-root, about 1/3 of the probes improve latency by 50ms; for K-root, 23% do. The line for D-root shows a "step" behavior because for some probes, the geo-hint helps avoid cross-continental or cross-oceanic links K-root has a lot more global sites than D-root, and the latency improvements are more evenly distributed.

There are "negative tails" in the results, which show that the geographic hint does harm anycast performance in rare cases. Only about 11 probes (0.7%) for C-

root, 201 probes (7.2%) for D-root and 83 (5.9%) for K-root are adversely affected. Among those, 2 probes to C-root received 20ms+ increased latency; 57 probes for D-root and 33 for K root. I further inspected the such probes and found in most cases that such negative effects of geographic hints are caused by bad links between the probe and geo-hinted, geographically nearby site.

The evaluation I conducted may underestimate the potential benefit of geographic hints. I obtain decision points from traceroute measurements to sampled sites, while an exhaustive probing could add new decision points that could expose a route to an better site. Also note that choosing the route towards the closest site may not lead to actually using that closest site. For example, consider a route that is advertised from an anycast site in Florida to an ISP in South America, and should be chosen as the route to the geographically closest site. However, it might be the case that the route traverses Virginia along the way, and the site in Virginia is actually the one with lowest latency for the clients in South America. In my evaluation, since I do not obtain traceroute towards Virginia site, the decision point does not have information about the Virginia site, thus use the Florida site as the geo-hinted site. This causes me to overestimate the latency to geo-hinted site, thus underestimate the benefit of geographic hints.

A concrete implementation of the geographic hint in BGP announcements would be embedding the information in BGP community tag. BGP community tags have the format X:Y, where X, Y are two 16-bit values. By convention, the first 16 bits are used to represent the AS number of the operator that sets the community. For the geo-hint, the last 16 bits can encode coarse latitude and longitude.

Latitude varies -90 to 90, but inhabited latitude is more -50 to 74 [127] and can thus be encoded in 7 bits. Longitude varies -180 to 180, so can be encoded in the remaining 9 bits easily. Note that encoding location information in BGP community tags is already used by many ISPs to record IXPs traversed by the BGP announcements [125] or where the route is received [122]. Anycast sites would include the community tags in out-bound advertisements, these tags would propagate as normal community tags do, and recipients would be allowed to choose to select routes considering the proximity of the destination(s) encoded in the last 16 bits.

This static "hint" scheme has little overhead to BGP announcements, since the geographic information is embedded in a 32-bits BGP community tag. When selecting routes, it is computationally light for BGP routers to evaluate the distances to anycast sites listed in the BGP announcements. More importantly, this scheme is incrementally deployable. For each router that recognizes and evaluates such geographic hints, it directs its traffic towards the geographically closest site to the router. The traffic through the router will benefit from the hint, regardless of how many other routers are configured to use the hint.

6.2 Other BGP hints

In the previous section, I evaluate how much benefit a static geographic hint would provide to anycast. There are other forms of extensions in BGP announcements, both static or dynamic, can be added specifically to anycast prefixes. For example, another static hint is the number of sites reachable via the route in the announce-

ment. Based on this number, BGP routers could choose the route that leads to the most sites, and expect the closest among the many sites will be closer. This hint is actually utilizing the idea of preferring the route leads to the common provider for the most anycast sites, a generalization of Ballani's suggestion of deploying anycast with a single provider [18]. This hint is represented simply with an integer, introduces minimal overhead to BGP announcements as well. But such a static hint may suffer when the closer sites are not included in the route towards most sites.

On the other hand, dynamic hints based on local measurements of load or latency through different routes, improve BGP's route selection mechanism for not only anycast prefixes but also unicast prefixes. But dynamic updates require extensive measurements and more sophisticated algorithms to evaluate performance of different routes.

The major advantage the proposed fixes, including the static hints, is that each of them is incrementally deployable and compatible with current BGP policy. If the hints are not recognized by some BGP router or even are removed, the performance will fall back to that of current anycast under default BGP policy. Moreover, each of the hints is flexible enough to allow additive usage in BGP announcements, and different ISPs could apply their own route selection mechanisms to evaluate hints independently.

6.3 Propagation of the proposed BGP hints

In this section, I conduct experiments to evaluate whether the proposed fixes in BGP community tags propagate well enough in the Internet to be used by distant ISPs. I implemented an experimental anycast deployment on PEERING testbed [27]. As described in Section 5.1.2, it allows researchers to announce prefixes from PEERING servers in different locations. With the same prefix announced from ten locations, the testbed implements an anycast deployment with ten global sites. Moreover, PEERING allows a researcher to announce prefixes with customized BGP community tags. For each server, which represents an anycast site, I embed specific code (e.g., hash of the location city name) for it in the BGP community tags it announces. I announce the prefix allocated to PEERING 184.164.249.0/24 (ASN47065) from ten different locations including Amsterdam, Athens, Belo Horizonte (Brazil), Boston, Clemson, Los Angeles, Phoenix, Salt Lake City and Seattle.

I characterize the propagation of community tags in BGP announcements by collecting BGP routes towards the prefix announced from PEERING from 20 Route-Views [128] BGP collectors. Among the 20 collectors, 11 of them received routes with customized community tags from at least one route towards the prefix. The fraction of routes to our announced prefix that have the community tags ranges from 8% to 38% on the 11 collectors. Five collectors received tags from their closest sites, i.e., they are presented with the routes to their closest sites. The other 5 received tags from their second closest replicas; another one is provided with tags to the fourth closest replica. Note that by default, Cisco routers do not pass BGP

community tags to their peers [129]. The results from this experiment are encouraging: Many of the clients benefit from the geo-hints even with the BGP community filtering as in today's Internet.

To understand if the customized community tags from PEERING are treated differently from BGP communities that are already used in practice, I further characterize the propagation of community tags from other ISPs, including ServerCentral [130], Packet Clearing House [131] and Init7 [132]. I find similar propagation of BGP communities from the measured ISPs as from PEERING testbed: 7 to 13 collectors received routes with community tags, and usually less than 50% of routes received at the collectors contain community tags.

Recent work [115] on analyzing BGP community propagation confirms our observations. Streibelt et al. [115] evaluated BGP community propagation by collecting passive route collections from RouteViews and RIPE RIS, etc., and active measurements from looking glasses and RIPE Atlas probes. They reported that over 75% of all BGP announcements collected at more than 190 BGP collectors have at least one community tag. Moreover, they found that over 50% of the communities are propagated over four AS hops, so that the majority of communities are propagated through the entire Internet.

6.4 Concerns on BGP community tags

BGP communities are widely used mechanism by network operators to manage policy, engineer traffic or mitigate attacks. Results in this chapter show that a

simple static hint in BGP community tags could significantly reduce anycast path inflation. Unfortunately, BGP communities can also be exploited by adversarial parties to influence routing in malicious ways.

Streibelt et al. [115] demonstrate with experiments in the real Internet that many BGP community-based attacks, including remotely triggered blackholing, traffic steering and route manipulation of prefixes from another ISP, are easy to achieve. These attacks are feasible mainly due to the several weakness in current use and implementation of BGP communities and community-based policies. First, BGP communities effectively propagate in the entire Internet, as shown in my last section and Streibelt's work [115]. While their propagation allows network operators to implement routing policies with additional information, it provides opportunities to attackers to launch community-based attacks remotely, with or without hijacking the target prefixes. Second, it is extremely concerning that inserting or modifying BGP community tags in announcements requires no permission or authentication. In general, the adoption of authentication in Internet protocols has been shown to be a slow process. In addition, many BGP communities can be setup locally by each ISPs and have no standardized semantic. As such, communities used for one ISP may cause unintentional interference for others. The last but not least, there lacks effective monitoring of the usage of BGP communities, and thus leaves the abuse of community tags to grow.

The Internet community should take steps to address the major weakness in the use of BGP community tags. To allow better use of BGP communities, especially those which are purely informative like the one proposed in this dissertation,

the network operational community should standardize and publish well-known and tested BGP communities semantics and best practice configurations. More importantly, cryptographic integrity and authenticity for BGP communities need to be ensured in community-based services and policies. Current cryptographic mechanisms to protect the integrity and authenticity of routing announcement do not cover BGP communities [133–136]. As BGP communities are increasingly popular and widely used to embed various information, it is important to provide security check mechanisms for them.

6.5 Conclusion

Prior studies provide surprisingly few suggestions on improving anycast performance. Ballani et al. [18, 24] suggested deploy all anycast instances such that they all share the same upstream provider. However, this solution requires extensive of deployment changes for services that are not working under the deployment. Moreover, it introduces a single point of failure, the common upstream provider, in anycast, and makes anycast more vulnerable to interruptions in the provider.

In this chapter, I propose a fix to anycast performance deficit by adding static geographic hints in its BGP announcements. BGP routers are able to identify the routes going to the geographically closest anycast site through such static hints. ISPs obtain estimates on performance of routes, and can thus tie-break more intelligently among routes to different anycast sites. I evaluate the benefits of such static hints by simulating the route selection process with real traces. The results show that this

fix significantly reduces latency inflation for almost all clients, and reduces latency by 50 ms for 23% to 33% of the clients of D-root and K-root. This proposed fix has several advantages: it is incrementally deployable; it does not require deployments changes; and it has little operation overhead.

To analyze the effectiveness of the fix in real Internet, I further evaluate the propagation of the static hints embedded in BGP communities using PEERING testbed. My results mostly reaffirm the observations by Streibelt et al. [115]: Most ISPs forward BGP community tags they receive onward, and thus most BGP communities are propagated globally. In the end, I discuss major weakness of BGP communities and their usage in current Internet, and provide suggestions to the Internet operators and research community to improve effectiveness and security in BGP community-based services and operations.

Chapter 7: Conclusion

In this dissertation, I describe measurement-based methods to diagnose Internet anycast, and provide fixes to improve its performance. While prior studies found anycast can be inefficient, they provided little quantification on how bad the inefficiencies are and how common it is for anycast deployments to experience the performance problems. In Chapter 3, I provide a comprehensive and longitudinal performance measurements of distinct anycast deployments. By measuring nine root DNS servers and three major open DNS resolvers, I conclude that anycast, in most deployments, is neither effective at directing queries to nearby replicas, nor does it distribute traffic in a balanced manner. Furthermore, with longitudinal measurement over 2 years, I show that performance of the most anycast deployments is not improved over the time, even with dozens of new sites added for some of them.

To investigate the root causes for inefficiencies in anycast, in Chapter 4, I develop novel measurement techniques to quantify the anycast and unicast path inflation. I analyze the route selection process based on the standard BGP model to understand what make ISPs to choose the routes to distant anycast sites. It turns out much of the anycast inflation is due to ISPs do not have information to tie-break more intelligently among seemingly equal routes. Although the performance-

awareness is a known weakness of BGP, I quantitatively show that it is amplified and causes more path inflation in anycast scenarios.

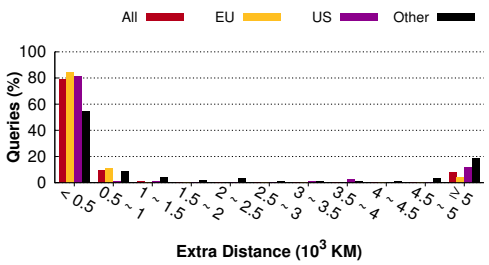
Based on the insights from Chapter 4, i.e., ISPs' poor tie-breaking causes much of the anycast inflation, I further investigate what are the route preferences they have when tie-breaking. In Chapter 5, I design control plane experiments and use large scale measurements to directly identify relative preferences among routes. My results show that in about 30% of the cases ISPs do not tie-break randomly or with "Shortest AS path", but have preferences towards routes from certain providers. I show that many Tier-2 ISPs have common and consistent preferences among Tier-1 ISPs, even when they have the same provider-customer relations to most ISPs. The newly discovered preferences provide useful complement to Gao-Rexford model on BGP routing, and can help research community in various BGP-related studies. Also, our results indicate that about 50% of the Tier-2 ISPs do not choose shorter paths from Tier-1 providers even when available. This observation suggests researchers and network operators to re-evaluate the effectiveness of selective AS path prepending as a traffic engineering method. Fortunately, I show that newly discovered preferences do not cause additional deficits in anycast performance.

In Chapter 6, I provide suggestions to the Internet community on improving anycast performance. I describe an incrementally deployable fix to the inefficiency of anycast. With simulation on real network traces, I show that the fix can reduce latency inflation for almost all clients. Overall, the proposed fix and all previous results provide comprehensive understanding on anycast in today's Internet. The contributions in this dissertation help improve performance and reliability of critical

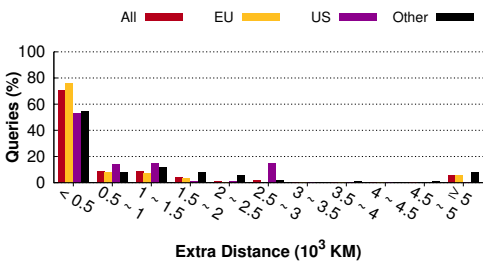
Internet infrastructure, and benefit global Internet users.

Chapter 8: Appendix

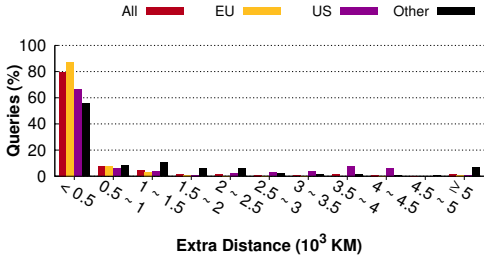
In this appendix, I show the figures that illustrate the extra distance measure for all nine measured DNS roots. The figures are generated the same way as in Figure 3.5 and 3.6.



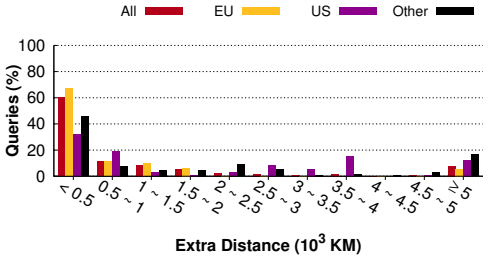
(a) A-root



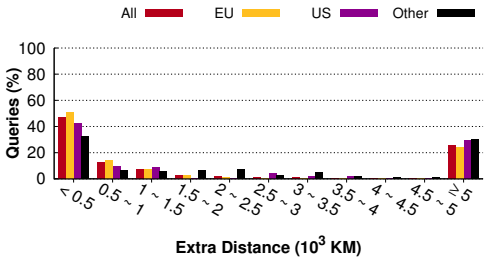
(b) E-root



(c) F-root

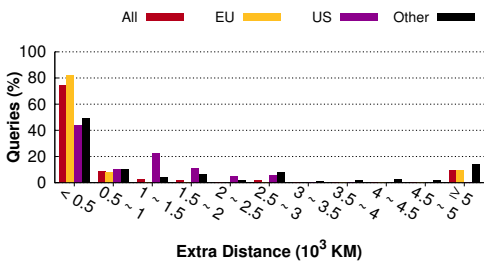


(d) I-root

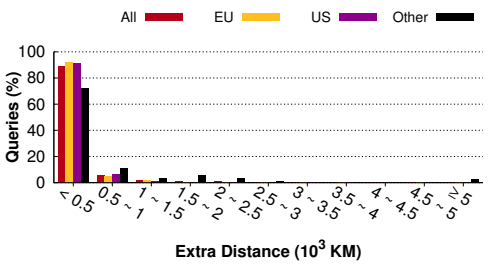


(e) J-root

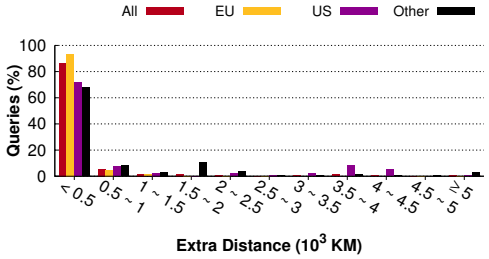
Figure 8.1: Distribution of RIPE Atlas queries to various DNS roots over additional distance (compared to their closest sites) traveled in December 2017.



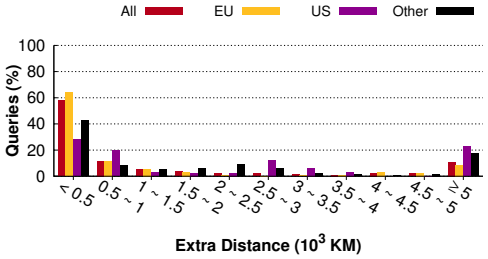
(a) A-root



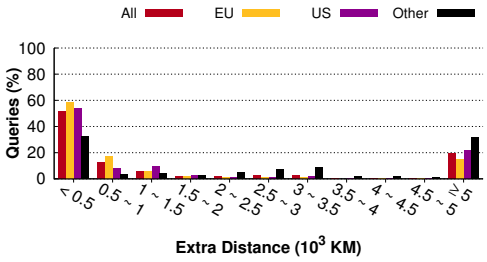
(b) E-root



(c) F-root



(d) I-root



(e) J-root

Figure 8.2: Distribution of RIPE Atlas queries to various DNS roots over additional distance (compared to their closest sites) traveled in March 2019.

Bibliography

[1] Voipfone. Voip 999 emergency services. http://www.voipfone.co.uk/999_Emergency_Services.php.

[2] Federal Communications Commission. Voip and 911 service. https://www.fcc.gov/consumers/guides/voip-and-911-service.

[3] Google Public DNS. Frequently asked questions. https://developers.google.com/speed/public-dns/faq#locations.

[4] OpenDNS. OpenDNS data center locations. https://www.opendns.com/data-center-locations/.

[5] Cloudflare. The Cloudflare global anycast network. https://www.cloudflare.com/network/.

[6] Giovane Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Muller, Lan Wei, and Cristian Hesselman. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 255–270. ACM, 2016.

[7] Ricardo de Oliveira Schmidt, John Heidemann, and Jan Harm Kuipers. Anycast latency: How many sites are enough? In *Passive and Active Network Measurement Workshop (PAM)*, pages 188–200. Springer, 2017.

[8] Daniel Karrenberg. Anycast and BGP stability: a closer look at DNSMon (talk). http://meetings.ripe.net/ripe-50/presentations/ripe50-plenary-tue-anycast.pdf, 2005.

[9] Lorenzo Colitti, Erik Romijn, Henk Uijterwaal, and Andrei Robachevsky. Evaluating the effects of anycast on DNS Root name servers. *RIPE document RIPE-393*, 6, 2006.

[10] Cloudflare, Inc. Delivering Dot. https://blog.cloudflare.com/f-root/.

[11] Wouter B. de Vries, Ricardo de O. Schmidt, Wes Hardaker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. Broad and load-aware anycast mapping with Verfploeter. In *ACM Internet Measurement Conference (IMC)*, pages 477–488. ACM, 2017.

[12] Christopher Metz. IP anycast point-to-(any) point communication. *IEEE Internet computing*, 6:94–98, 2002.

[13] Cloudflare, Inc. Announcing 1.1.1.1: the Fastest, Privacy-First Consumer DNS Service. `https://blog.cloudflare.com/announcing-1111/`.

[14] Cloudflare, Inc. Fixing Reachability to 1.1.1.1, GLOBALLY! `https://blog.cloudflare.com/fixing-reachability-to-1-1-1-1-globally/`.

[15] Root-servers.org. Root Servers Archives. `http://root-servers.org/archives/`.

[16] Sandeep Sarat, Vasileios Pappas, and Andreas Terzis. On the use of anycast in DNS. In *Computer Communications and Networks, 2006. ICCCN 2006. Proceedings. 15th International Conference on*, pages 71–78. IEEE, 2006.

[17] Jinjin Liang, Jian Jiang, Haixin Duan, Kang Li, and Jianping Wu. Measuring query latency of top level DNS servers. In *Passive and Active Network Measurement Workshop (PAM)*, pages 145–154. Springer, 2013.

[18] Hitesh Ballani, Paul Francis, and Sylvia Ratnasamy. A measurement-based deployment proposal for IP anycast. In *ACM Internet Measurement Conference (IMC)*, 2006.

[19] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the performance of an Anycast CDN. In *ACM Internet Measurement Conference (IMC)*, pages 531–537. ACM, 2015.

[20] Ziqian Liu, Bradley Huffaker, Marina Fomenkov, Nevil Brownlee, et al. Two days in the life of the DNS anycast root servers. In *Passive and Active Network Measurement Workshop (PAM)*, pages 125–134. Springer, 2007.

[21] Jan Harm Kuipers. Analyzing the K-root DNS anycast infrastructure. *Twente Student Conference on IT*, 2015.

[22] Biet Barber, Matt Larson, and Mark Kosters. Traffic source analysis of the J-root anycast instances (Talk). `https://www.nanog.org/meetings/nanog39/presentations/larson.pdf`, 2006.

[23] Lan Wei and John Heidemann. Does anycast hang up on you? *IEEE Transactions on Network and Service Management*, 2018.

[24] Hitesh Ballani and Paul Francis. Towards a global IP anycast service. In *ACM SIGCOMM*, 2005.

[25] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM Transactions on Networking (TON)*, 9(6):681–692, 2001.

[26] Lixin Gao, Timothy G. Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, volume 1, pages 547–556. IEEE, 2001.

[27] Brandon Schlinker, Kyriakos Zarifis, Italo Cunha, Nick Feamster, and Ethan Katz-Bassett. Peering: An AS for us. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, page 18. ACM, 2014.

[28] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[29] RIPE NCC. Test traffic measurements service. `http://www.ripe.net/ttm/`.

[30] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating latency between arbitrary Internet end hosts. In *ACM Internet Measurement Workshop (IMW)*, 2002.

[31] Bu-Sung Lee, Yu Shyang Tan, Yuji Sekiya, Atsushi Narishige, and Susumu Date. Availability and effectiveness of Root DNS servers: A long term study. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 862–865. IEEE, 2010.

[32] Sharad Agarwal, Chen-Nee Chuah, Supratik Bhattacharyya, and Christophe Diot. The impact of BGP dynamics on intra-domain traffic. In *ACM SIGMETRICS Performance Evaluation Review*, volume 32, pages 319–330. ACM, 2004.

[33] Nick Feamster, David G Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of internet path faults on reactive routing. In *ACM SIGMETRICS Performance Evaluation Review*, volume 31, pages 126–137. ACM, 2003.

[34] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed internet routing convergence. *ACM SIGCOMM Computer Communication Review*, 30:175–187, 2000.

[35] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental study of internet stability and backbone failures. In *Fault-Tolerant Computing, 1999. Digest of Papers. Twenty-Ninth Annual International Symposium on*, pages 278–285. IEEE, 1999.

[36] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, and Christophe Diot. Characterization of failures in an IP backbone. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2307–2317. IEEE, 2004.

[37] Matthew Roughan, Tim Griffin, Morley Mao, Albert Greenberg, and Brian Freeman. Combining routing and traffic data for detection of IP forwarding anomalies. *ACM SIGMETRICS Performance Evaluation Review*, 32:416–417, 2004.

[38] Renata Teixeira, Aman Shaikh, Tim Griffin, and Jennifer Rexford. Dynamics of hot-potato routing in IP networks. In *ACM SIGMETRICS Performance Evaluation Review*, volume 32, pages 307–319. ACM, 2004.

[39] Vern Paxson. End-to-end routing behavior in the Internet. *IEEE/ACM transactions on Networking*, 5:601–615, 1997.

[40] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A measurement study on the impact of routing events on end-to-end Internet path performance. In *ACM SIGCOMM Computer Communication Review (CCR)*, volume 36, pages 375–386. ACM, 2006.

[41] Himabindu Pucha, Ying Zhang, Z Morley Mao, and Y Charlie Hu. Understanding network delay changes caused by routing events. In *ACM SIGMETRICS performance evaluation review*, volume 35, pages 73–84. ACM, 2007.

[42] Yaron Schwartz, Yuval Shavitt, and Udi Weinsberg. On the diversity, stability and symmetry of end-to-end Internet routes. In *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pages 1–6. IEEE, 2010.

[43] Yaron Schwartz, Yuval Shavitt, and Udi Weinsberg. A measurement study of the origins of end-to-end delay variations. In *International Conference on Passive and Active Network Measurement*, pages 21–30. Springer, 2010.

[44] Udi Weinsberg, Yuval Shavitt, and Yaron Schwartz. Stability and symmetry of Internet routing. In *INFOCOM Workshops 2009, IEEE*, pages 1–2. IEEE, 2009.

[45] Ying Zhang, Zhuoqing Morley Mao, and Jia Wang. A framework for measuring and predicting the impact of routing changes. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 339–347. IEEE, 2007.

[46] Ying Zhang, Zhuoqing Morley Mao, and Ming Zhang. Effective Diagnosis of Routing Disruptions from End Systems. In *Symposium on Networked Systems Design and Implementation (NSDI)*, volume 8, pages 219–232, 2008.

[47] Italo Cunha, Renata Teixeira, Darryl Veitch, and Christophe Diot. Predicting and tracking internet path changes. In *ACM SIGCOMM Computer Communication Review (CCR)*, volume 41, pages 122–133. ACM, 2011.

[48] Kyriaki Levanti, Sihyung Lee, and Hyong S. Kim. On reducing the impact of interdomain route changes. In *Passive and Active Network Measurement Workshop (PAM)*, pages 153–162. Springer, 2011.

[49] Massimo Rimondini, Claudio Squarcella, and Giuseppe Di Battista. Towards an automated investigation of the impact of BGP routing changes on network delay variations. In *International Conference on Passive and Active Network Measurement*, pages 193–203. Springer, 2014.

[50] Giordano Da Lozzo, Giuseppe Di Battista, and Claudio Squarcella. Visual discovery of the correlation between BGP routing and round-trip delay active measurements. *Computing*, 96:67–77, 2014.

[51] Romain Fontugne, Johan Mazel, and Kensuke Fukuda. An empirical mixture model for large-scale RTT measurements. In *INFOCOM, 2015 Proceedings IEEE*, pages 2470–2478. IEEE, 2015.

[52] Wenqin Shao, Jean-Louis Rougier, Antoine Paris, François Devienne, and Mateusz Viste. One-to-One matching of RTT and path changes. In *Teletraffic Congress (ITC 29)*, volume 1, pages 196–204. IEEE, 2017.

[53] James Hiebert, Peter Boothe, Randy Bush, and Lucy Lynch. Determining the cause and frequency of routing instability with anycast. In *Asian Internet Engineering Conference*, pages 172–185. Springer, 2006.

[54] Peter Boothe and Randy Bush. DNS anycast stability. *19th APNIC,05*, 2005.

[55] Ray Bellis. Researching F-root anycast placement using RIPE Atlas, 2015.

[56] Michael J Freedman, Karthik Lakshminarayanan, and David Mazières. OASIS: Anycast for any service. In *NSDI*, volume 6, pages 10–10, 2006.

[57] Zakaria Al-Qudah, Seungjoon Lee, Michael Rabinovich, Oliver Spatscheck, and Jacobus Van der Merwe. Anycast-aware transport for Content Delivery Networks. In *Proceedings of the 18th international conference on World wide web*, pages 301–310. ACM, 2009.

[58] Hussein A Alzoubi, Seungjoon Lee, Michael Rabinovich, Oliver Spatscheck, and Jacobus Van Der Merwe. A practical architecture for an anycast CDN. *ACM Transactions on the Web (TWEB)*, 5:17, 2011.

[59] Geoff Huston. Peering and settlements - part 1. *The Internet Protocol Journal (Cisco)*, 1999.

[60] Geoff Huston. Peering and settlements - part 2. *The Internet Protocol Journal (Cisco)*, 1999.

[61] Timothy G Griffin and Gordon Wilfong. An analysis of bgp convergence properties. *ACM SIGCOMM Computer Communication Review*, 29(4):277–288, 1999.

[62] Timothy G Griffin, F Bruce Shepherd, and Gordon Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking (ToN)*, 10(2):232–243, 2002.

[63] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. Internet routing resilience to failures: analysis and implications. In *Proceedings of the 2007 ACM CoNEXT conference*, page 25. ACM, 2007.

[64] Umar Javed, Italo Cunha, David Choffnes, Ethan Katz-Bassett, Thomas Anderson, and Arvind Krishnamurthy. Poiroot: Investigating the root cause of interdomain path changes. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 183–194. ACM, 2013.

[65] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Nation-state routing: Censorship, wiretapping, and BGP. *arXiv preprint arXiv:0903.3218*, 2009.

[66] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. RAPTOR: Routing attacks on privacy in Tor. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 271–286, 2015.

[67] Yixin Sun, Anne Edmundson, Nick Feamster, Mung Chiang, and Prateek Mittal. Counter-RAPTOR: Safeguarding Tor against active routing attacks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 977–992. IEEE, 2017.

[68] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Internet anycast: performance, problems, & potential. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 59–73. ACM, 2018.

[69] Haowen Chan, Debabrata Dash, Adrian Perrig, and Hui Zhang. Modeling adoptability of secure BGP protocol. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 279–290. ACM, 2006.

[70] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. *ACM SIGCOMM computer communication review*, 41(4):14–25, 2011.

[71] Sharon Goldberg, Michael Schapira, Peter Hummon, and Jennifer Rexford. How secure are secure interdomain routing protocols. *ACM SIGCOMM Computer Communication Review*, 41:87–98, 2011.

[72] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Autonomous security for autonomous systems. *Computer Networks*, 52:2908–2923, 2008.

[73] Maciej Wojciechowski, Benno Overeinder, Guillaume Pierre, Maarten Van Steen, and Janina Mincer-daszkiewicz. Border gateway protocol modeling and simulation. *Master thesis*, 2008.

[74] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Modeling on quicksand: Dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM Computer Communication Review*, 42(1):40–46, 2012.

[75] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. Inferring complex AS relationships. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 23–30. ACM, 2014.

[76] Harsha V Madhyastha, Ethan Katz-Bassett, Thomas E Anderson, Arvind Krishnamurthy, and Arun Ve nkataramani. iplane nano: Path prediction for peer-to-peer applications. In *NSDI*, volume 9, pages 137–152, 2009.

[77] Wolfgang Mühlbauer, Anja Feldmann, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Building an as-topology model that captures route diversity. *ACM SIGCOMM Computer Communication Review*, 36:195–206, 2006.

[78] Phillipa Gill, Michael Schapira, and Sharon Goldberg. A survey of interdomain routing policies. *Computer Communication Review*, 44(1):28–34, 2014.

[79] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. Investigating interdomain routing policies in the wild. In *Proceedings of the 2015 Internet Measurement Conference*, pages 71–77. ACM, 2015.

[80] Lorenzo Colitti. Internet topology discovery using active probing. *Ph.D. thesis*, 2006.

[81] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet optometry: assessing the broken glasses in internet reachability. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*, pages 242–253. ACM, 2009.

[82] Ethan Katz-Bassett, Colin Scott, David R Choffnes, Ítalo Cunha, Vytau tas Valancius, Nick Feamster, Harsha V Madhyastha, Thomas Anderson, and Arvind Krishnamurthy. Lifeguard: Practical repair of persistent route failures. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 395–406. ACM, 2012.

[83] Muoi Tran, Min Suk Kang, Hsu-Chun Hsiao, Wei-Hsuan Chiang, Shu-Po Tung, and Yu Su Wang. On the feasibility of rerouting-based DDoS defenses.

In *To appear in Proceedings of IEEE Symposium on Security and Privacy (IEEE S&P)*, 2019.

[84] Jared M Smith and Max Schuchard. Routing around congestion: Defeating ddos attacks and adverse network conditions via reactive bgp rou ting. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 599–617. IEEE, 2018.

[85] Matt Weinberg and Duane Wessels. Review and analysis of anonmalous traffic to A-root and J-root (Nov/Dec 2015). DNS-OARC 24 Presentation, 2016.

[86] OpenDNS. OpenDNS diagnostic tool. `https://support.opendns.com/hc/en-us/articles/227988487-Diagnostic-Tool-Link-and-Instructions`.

[87] RIPE NCC. RIPE Atlas. `https://atlas.ripe.net/`.

[88] RIPE NCC. RIPE Atlas: A global internet measurement network. *Internet Protocol Journal*, 18(3), 2015.

[89] MaxMind Inc. Maxmind geoip2 city. `https://www.maxmind.com/en/geoip2-databases`, 2017.

[90] Sharad Agarwal and Jacob R. Lorch. Matchmaking for Online Games and Other Latency-Sensitive P2P Systems. In *ACM SIGCOMM*, 2009.

[91] RIPE NCC. RIPE Atlas probes locations. `https://atlas.ripe.net/probes/`, 2017.

[92] Neil Spring, Ratul Mahajan, and Thomas Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, 2003.

[93] Center for Applied Internet Data Analysis (CAIDA). AS relationships dataset. `http://www.caida.org/data/as-relationships/`.

[94] Chiara Orsini, Alistair King, Danilo Giordano, Vasileios Giotsas, and Alberto Dainotti. BGPStream: a software framework for live and historical BGP data analysis. In *ACM Internet Measurement Conference (IMC)*, pages 429–444. ACM, 2016.

[95] Dyn Research. A bakers dozen, 2016 edition. `http://dyn.com/blog/a-bakers-dozen-2016-edition/`.

[96] Packet Clearing House (PCH). D-Root peering policy. `https://www.pch.net/services/dns_anycast`.

[97] Packet Clearing House (PCH). PCH Daily routing snapshots. `https://www.pch.net/resources/Routing_Data/`.

[98] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H Katz. Towards an accurate AS-level traceroute tool. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378. ACM, 2003.

[99] George Nomikos and Xenofontas Dimitropoulos. traIXroute: Detecting IXPs in traceroute paths. In *Passive and Active Network Measurement Workshop (PAM)*, pages 346–358. Springer, 2016.

[100] PeeringDB. PeeringDB exchanges. `https://www.peeringdb.com/`.

[101] Packet Clearing House (PCH). PCH Internet exchange directory. `https://www.pch.net/ixp/dir`.

[102] INAP Inc. InterNAP managed Internet route optimizer. `http://www.inap.com/network-services/miro-controller/`, 2017.

[103] Yi-Ching Chiu, Brandon Schlinker, Abhishek Balaji Radhakrishnan, Ethan Katz-Bassett, and Ramesh Govindan. Are we one hop away from a better internet? In *Proceedings of the 2015 Internet Measurement Conference*, pages 523–529. ACM, 2015.

[104] Mirjam K̈hne. Interesting graph - as path lengths over time. `https://labs.ripe.net/Members/mirjam/interesting-graph-as-path-lengths`.

[105] Cun Wang, Zhengmin Li, Xiaohong Huang, and Pei Zhang. Inferring the average as path length of the internet. In *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pages 391–395. IEEE, 2016.

[106] PEERING: The BGP testbed. `https://peering.usc.edu/`.

[107] The Internet2 Community. The Internet2. `https://www.internet2.edu/`, 2018.

[108] GEANT. GEANT website. `https://www.geant.org/`.

[109] The Internet2 Community. The Internet2. `https://www.internet2.edu/about-us/membership/`, 2018.

[110] GEANT. GEANT: About our membership. `https://www.geant.org/About/Membership`.

[111] PEERING acceptable use policy. `https://peering.usc.edu/aup//`.

[112] Jared M Smith, Kyle Birkeland, and Max Schuchard. An internet-scale feasibility study of BGP poisoning as a security primitive. *arXiv preprint arXiv:1811.03716*, 2018.

[113] BGP communities. `https://onestep.net/communities/`.

[114] RIPE whois database query. https://www.ripe.net/manage-ips-and-asns/db/support/querying-the-ripe-database.

[115] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. BGP communities: Even more worms in the routing can. In *Proceedings of the Internet Measurement Conference 2018*, pages 279–292. ACM, 2018.

[116] PEERING Testbed peering sessions. https://peering.usc.edu/peers/.

[117] BGP looking glass database. http://www.bgplookingglass.com/.

[118] BGP IPv4 route servers. https://www.bgp4.net/doku.php?id=tools:ipv4_route_servers.

[119] Ravi Chandra, Paul Traina, and Tony Li. BGP communities attribute. IETF RFC 1997, 1996.

[120] Srihari Sangli, Daniel Tappan, and Yakov Rekhter. BGP extended communities attribute. IETF RFC 4360, 2006.

[121] Bruno Quoitin, Cristel Pelsser, Louis Swinnen, Ouvier Bonaventure, and Steve Uhlig. Interdomain traffic engineering with BGP. *IEEE Communications magazine*, 41:122–128, 2003.

[122] Benoit Donnet and Olivier Bonaventure. On BGP communities. *ACM SIGCOMM Computer Communication Review*, 38:55–59, 2008.

[123] Christoph Dietzel, Anja Feldmann, and Thomas King. Blackholing at IXPs: On the effectiveness of DDoS mitigation in the wild. In *International Conference on Passive and Active Network Measurement*, pages 319–332. Springer, 2016.

[124] K. Foster. Application of BGP communities. *The Internet Protocol Journal*, 6:2–9, 2003.

[125] Vasileios Giotsas, Christoph Dietzel, Georgios Smaragdakis, Anja Feldmann, Arthur Berger, and Emile Aben. Detecting peering infrastructure outages in the wild. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 446–459. ACM, 2017.

[126] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 133–145. ACM, 2002.

[127] Radical Cartography. World's Population in 2000, by Latitude. http://www.radicalcartography.net/index.html?histpop, 2017.

[128] University of Oregon. Route Views project. http://www.routeviews.org/.

[129] Cisco Systems, Inc. Cisco 'send-community' command. https://www.cisco.com/c/m/en_us/techdoc/dc/reference/cli/n5k/commands/send-community.html.

[130] ServerCentral Management. ServerCentral BGP communities. https://www.servercentral.com/bgp-communities/.

[131] Packet Clearing House (PCH). Peering with Packet Clearing House. https://www.pch.net/about/peering.

[132] Init7 NOC. BGP communities for Init7 customers. https://as13030.net/static/pdf/as13030_bgp_communities.pdf.

[133] Randy Bush and Rob Austein. The resource public key infrastructure (RPKI) to router protocol. Technical report, IETF, January 2013. RFC 6810.

[134] Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick D. McDaniel, and Aviel D. Rubin. Working around BGP: An incremental approach to improving security and accuracy in interdomain routing. In *NDSS*, volume 23, page 156, 2003.

[135] Ethan Heilman, Danny Cooper, Leonid Reyzin, and Sharon Goldberg. From the consent of the routed: Improving the transparency of the RPKI. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 51–62. ACM, 2014.

[136] Yih-Chun Hu, Adrian Perrig, and Marvin Sirbu. SPV: Secure path vector routing for securing BGP. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 179–192. ACM, 2004.