Internet Anycast: Performance, Problems, & Potential

Zhihao Li
University of Maryland

Dave Levin
University of Maryland

Neil Spring
University of Maryland

Bobby Bhattacharjee
University of Maryland

ABSTRACT

Internet anycast depends on inter-domain routing to direct clients to their “closest” sites. Using data collected from a root DNS server for over a year (400M+ queries/day from 100+ sites), we characterize the load balancing and latency performance of global anycast. Our analysis shows that site loads are often unbalanced, and that most queries travel longer than necessary, many by over 5000 km.

Investigating the root causes of these inefficiencies, we can attribute path inflation to two causes. Like unicast, anycast routes are subject to inter-domain routing topology and policies that can increase path length compared to theoretical shortest (e.g., great-circle distance). Unlike unicast, anycast routes are also affected by poor route selection when paths to multiple sites are available, subjecting anycast routes to an additional, unnecessary, penalty.

Unfortunately, BGP provides no information about the number or goodness of reachable anycast sites. We propose an additional hint in BGP advertisements for anycast routes that can enable ISPs to make better choices when multiple “equally good” routes are available. Our results show that use of such routing hints can eliminate much of the anycast path inflation, enabling anycast to approach the performance of unicast routing.

ACM Reference Format:
Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2018. Internet Anycast: Performance, Problems, & Potential. In *Proceedings of ACM SIGCOMM (SIGCOMM).* ACM, New York, NY, USA, 15 pages. https://doi.org/

1 INTRODUCTION

Anycast is one of the fundamental modes of communication, in which a set of anycast *replicas* all serve the same content under a shared identifier. In IP anycast in particular, server *replicas* at multiple geographic *sites* advertise the same IP address via BGP; clients are “routed” to a replica based on the underlying BGP routes; and from a client’s perspective, all of the anycast replicas offer an equivalent service [23, 31, 39]. This basic *one-to-any* form of communication is used in critical network infrastructure: all root DNS servers and many popular open resolvers are hosted via IP anycast [11, 12, 26, 44], and some content delivery networks (CDNs) use it in an attempt to lower latencies and distribute load [7].

What makes IP anycast attractive when deploying a globally replicated service is the mental model that it seems to permit. In particular, as one adds more anycast replicas in locations with many clients, it is generally believed [3, 7, 47] that: (1) overall client latency will decrease and (2) load from nearby clients will be more evenly distributed. Of course, inter-domain routing is not guaranteed to be optimal in terms of bandwidth, latency, or geographic proximity: at best, BGP can be relied upon for connectivity and policy-compliance. Nonetheless, as evidenced by the increasing deployment of anycast in root DNS servers and open resolvers, network operators expect these broad trends, at least, to apply.

Unfortunately, several prior studies have found that IP anycast’s performance does not match even these most basic expectations. Clients are often routed to replicas that are hundreds of kilometers away from their closest replicas [24, 28], resulting in increased latency. It has been known for over a decade that IP anycast can be inefficient, and yet there are surprisingly few explanations of *why* or *how to fix it.*

To the best of our knowledge, the only concrete proposed solution comes from Ballani et al. [3, 4], who hypothesized that deploying anycast replicas such that they all share the same upstream provider is one approach to fix inefficiencies in anycast. Were this the only solution, it would mean that efficiently running a geo-replicated service over anycast would require cooperation from a large ISP; adding even a small ISP could negatively impact performance. Is this a fundamental limitation of anycast, or is there another solution?

In this paper, we present an in-depth analysis of three distinct IP anycast deployments: those of the C-, D-, and K-root DNS servers.[1] We investigate the current inefficiencies

SIGCOMM, August 2018, Budapest, Hungary

[1] We have also analyzed other root DNS servers, and have found them to be largely similar to the three we focus on; we omit them due to space.

of IP anycast, why it fails (and succeeds), and how to fix it without relying upon a single large upstream provider. This paper makes three broad, interrelated contributions:

Performance (§3): Using passive and active measurements of distinct, root DNS anycast deployments, we quantify the inefficiencies of IP anycast in terms of both latency and load balance. While it is not surprising that IP anycast is suboptimal (BGP routing lacks mechanisms to select routes with better performance), we find the inefficiencies to be surprisingly excessive. In particular, we show that adding more anycast replicas (1) often *increases* overall latency, and (2) often exacerbates load balancing, matching clients to anycast replicas in different continents than their own.

Problems (§4): To explore the root causes of these performance problems, we introduce a novel measurement technique that allows us to compare the AS-level paths from clients to multiple IP anycast sites. The resulting data indicate that the majority of performance inefficiency is due to BGP's poor route selection: routers are presented with routes to two or more anycast replicas each of whom have equal preference. Lacking any useful information to distinguish between them, routers often select a distant, high-latency anycast site over the closer, low-latency one. Again, it is not surprising that inter-domain routing would not choose the best alternative, but it is surprising that the best alternative is often an *unselected* option.

Potential (§5): Finally, applying our findings from our root-cause analysis, we propose a fix. We propose to include geographic hints in BGP advertisements that routers can use to more intelligently select routes among alternatives with equal preference. We find that this reduces the anycast path inflation (the additional latency imposed specifically by anycast) to *zero* for over 65% of clients. This technique is incrementally deployable and, although we evaluate it only on root DNS data and an experimental deployment, it can be applied to any IP anycast system.

Our results collectively provide an accurate, in-depth understanding of why IP anycast currently does not work, and how it can. To assist practitioners and researchers in better understanding and mitigating IP anycast's inefficiencies, our nonsensitive datasets are publicly available.[2]

2 RELATED WORK

IP anycast [23, 31, 39] is widely used to allows services to be transparently replicated across the Internet. Two of the most studied applications of IP anycast to date are root DNS servers [4, 13, 26, 27, 33, 45, 47] and content delivery networks (CDNs) [2, 6, 7, 14–16].

We organize our discussion of related work along the efforts of measuring, explaining, and fixing IP anycast performance. Although generally related to DNS performance, we focus here on work that studied root servers' use of IP anycast, and not more general studies of DNS server performance or availability [5, 38].

Performance measurements of IP anycast Several studies have compared the RTTs between clients and their anycast replicas to the *smallest* RTT among all of the possible anycast replicas [4, 13, 27, 45, 47].

Early studies of the performance of IP anycast among DNS root servers indicated a promising trend towards lowered latency. In 2006, Sarat et al. [45] performed an initial measurement of the additional latency induced by anycast on F- and K-root, and found that while few go to their lowest-latency replica, the latency overheads are typically small (75th percentiles of less than 5 msec for K-root and less than 20 msec for F-root). Colitti et al. [13] studied K-root in the same year and concluded with similar results. In 2010, Lee et al. [25] evaluated the loss rate and latency to root DNS servers that had deployed IP anycast. Their results from 2010 indicated a gradual decrease in the overall latencies from 2007 through most of 2008, followed by a gradual *increase* into the beginning of 2009. Given how early into the trend their study was, they were unable to account for the statistical significance or cause of this trend. Our study shows this trend to be real—IP anycast performance often decreases as more replicas are added—and identifies a root cause (BGP route selection) and a fix.

In 2013, Liang et al. [27] applied King technique [19] to measure latencies between about 20K open recursive resolvers and root DNS servers. Their results, however, showed that about 40% of the resolvers experienced latency overheads over 50 msec. Most recently, in 2016, Schmidt et al. [47] used RIPE atlas probes to measure RTTs to all DNS root servers that support anycast. They conclude that having "a few sites" is enough to achieve nearly as good performance as having many sites. Qualitatively, our results support this in the sense that adding many more sites does not improve performance, but we show that *this is a bug, not a feature*, in that many anycast deployments are unable to take advantage of performance that could be realized. In fact, we show that, for many anycast deployments, adding more replicas *harms* performance by increasing latency—a phenomenon originally predicted by Ballani et al. [4]. Like Schmidt et al. [47], we make use of RIPE atlas probes, and thus, also like them, are subject to the probes' Europe-centric bias. In §3.2, we demonstrate that this bias does not negatively impact our results.

Other studies have used the relative geographic distance as a metric for comparing how well anycast chooses among

[2] http://www.cs.umd.edu/projects/droot/

replicas. In 2006, Liu et al. [28] used two days' passive DNS data from C-, F-, K-root, and reported median additional distances (over the distances to their closest replicas) of 6000 km, 2000 km, and 2000 km, respectively. For C-root, they found that over 60% of clients traveled an extra 5000 km longer than strictly necessary; for F- and K-roots, 40% of clients traveled an extra 5000 km. Kuipers [24] performed a coarser-grained analysis of 10 minutes of K-root's anycast performance, showing that most clients are not getting routed to their geographically closest anycast replica. Our findings largely reinforce these prior results by showing that anycast can indeed be surprisingly far from optimal, but we expand them by identifying the root cause of these inefficiencies and by offering a fix.

Explaining and fixing IP anycast performance Many of the above measurement studies speculate that BGP routing has an impact on whether clients obtain their optimal replica, but have offered no concrete explanation or fix.

Sarat et al. [45] suggested that each anycast site has an announcement radius, and hypothesized that clients would select the route to topologically nearby site. However, change of advertisement radius does not ensure BGP to select the route towards the closer replica among the available routes. Ballani et al. [3, 4] hypothesized that IP anycast's latency inflation (what they refer to as the "stretch-factor") can be remedied by ensuring that all anycast replicas share a single upstream provider. Our study confirms this hypothesis; in particular, we find that C-root has such a deployment and does not suffer from the route selection problems that other root servers have. Most root servers are not deployed in this fashion; implementing this fix would require renegotiating their providers, a significant undertaking. Moreover, centralizing an anycast service's routing behind a large upstream provider introduces a single point of failure. Were this the only solution, it would mean that only very large ISPs could efficiently offer IP anycast; adding even a single small ISP could negatively impact performance. In §5, we introduce a more democratic fix: by adding static geographic hints to BGP, we can achieve nearly all of the same benefit as using a single upstream provider. In comparison to these prior proposals, our "geo-hints" are easily and immediately deployable, and they remain efficient even when there are many distinct upstream providers.

3 PERFORMANCE

We begin by studying the performance of Internet-wide anycast, using measurements of DNS root servers. The DNS root is served by 13 Internet addresses: A- through M-root. These addresses are administered by various different entities, and all root addresses are now served using anycast.

We use the following terminology: each address is anycast from different physical locations across the Internet, called *sites*. The same root address may be (and often is) anycast from different ASes. Each site may have multiple machines, called *replicas*. For a specific root, a given site is either *local* or *global*: replicas at local sites are available only within the AS in which they are located. Global replicas are advertised using inter-domain BGP, and can be accessed across the Internet. As of early 2018, some roots are anycast from hundreds of (global) sites, whereas others have fewer than ten [44]. In this paper, we consider each root to be a separate anycast service, and examine their behavior independently.

Fundamentally, we want to use our measurements to answer the following question: **Does anycast provide an intuitively *good* server selection mechanism?**

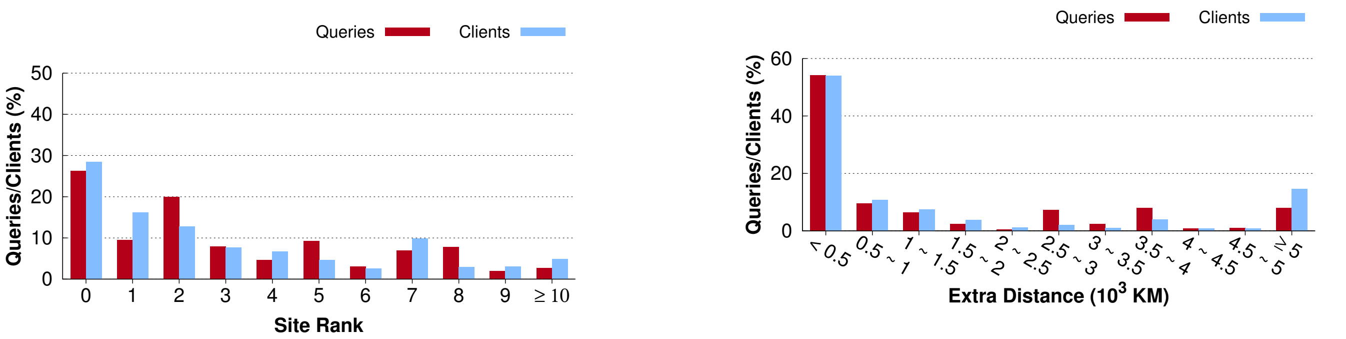
Server selection mechanisms may focus on various metrics. These include, but are not limited to, access latency, load balance, resilience, and geographic proximity. Our goal is to study whether anycast successfully improves these performance metrics.[3] In particular, we consider how these metrics improve as replicas are added.

We use two different sources of data in our analyses: traffic traces from the replicas of a root server, and active measurements from RIPE Atlas probes. We describe these datasets, including their features and limitations, next.

Root server traffic traces. Our first source of data is sampled traffic from the sites of D-root DNS server operated by University of Maryland. As of Jan 2018, D-root had over 120 anycast sites, 20 of which were global and the rest local [44]. We received 20% of all traffic at each replica, and base our analysis on data collected for every day in 2017. On average, in 2017, D-root received more than 30,000 queries per second, resulting in about 140 GB of trace data per day. This rich source of data allows us to understand client population and distribution that root servers see. This data also provides insight into load distribution, load variance, and inter-site traffic variation, each of which we analyze.

There are two limitations to the D-root dataset. First, it is data corresponding to a single root, and is subject to the policies of ASes that host sites. It is not clear if the performance for D-root extrapolates to anycast performance in general. Second, these data are entirely passively collected, and do not provide client-side latency measures or insight into alternate AS paths or other selection polices. To address

[3] We do not directly evaluate anycast resilience; however, we believe the dynamic hints described in §5 can be used to mitigate the effect of large-scale attacks like those that took place Nov. 30 and Dec. 1, 2015 [33, 52]. These attacks lasted for 2.5 hours on Nov. 30 and 1 hour on Dec. 1, resulting in a temporary take-down of B-, G-, and H-root, and increased response times from C-, E-, and K-root.



(a) D-root queries by rank

(b) Distribution of D-root queries by additional distance traveled.

Figure 1: D-root performance based on client traces.

both these problems, we augment this dataset with active measurements.

RIPE Atlas measurements. The RIPE Atlas framework [43] is a set of ~10,000 probes in 180 countries and ~3,587 ASes as of Jan 2018. Each probe periodically executes pre-defined measurements, called "build-in measurements", that include DNS CHAOS queries and traceroutes to all 13 DNS roots.

Our analysis uses queries that the RIPE Atlas probes sent to the 9 of 13 roots that have at least 5 anycast global sites [44]. DNS CHAOS queries retrieve data corresponding to the TXT record for the string "hostname.bind." with the DNS Class set to CHAOS (as opposed to Class Internet, which is the common case). The "hostname.bind." is a special record supported by BIND nameserver implementations, which is conventionally configured by the server operator to return a string that uniquely identifies the server replica.[4] These measurements allow us to record which specific replicas and sites a given probe (whose location is known [42]) is directed to by anycast over time. This specific type of DNS query was used in prior work, e.g., Moura et al. [33] and Schmidt et al. [47], to characterize anycast performance. We evaluate possible alternatives by augmenting this data with traceroutes and our own measurements of alternate replicas and addresses (§4).

3.1 How does anycast perform?

In this section, we characterize the performance of anycast service provided by D-root using our sampled traces.

Figures 1a and 1b show a measure of goodness of anycast for D-root. For each query received at D-root, we geo-locate the source of the query by IP address using the MaxMind database [30]. Next, we measure the distance from the query source to all D-root sites. For a query, the closest site is ranked 0, the next closest rank 1, and so on. We compute the same measure for each source IP address (client) as well.

We use geographic distance as an approximation of expected latency because the passive trace dataset taken at replicas does not provide a direct measure. Various studies have characterized the accuracy of MaxMind's geolocation [18, 20, 40, 49], by comparing with a sample of known locations or with a majority vote across databases. Although MaxMind may not be reliably precise to 10 km, these studies showed that it is within 300 km for approximately 80% of IP addresses. Our focus is on such coarse-grained geolocation, aggregating query distances in bins of 500 km, and the relatively small imprecision of the geolocation database is unlikely to be the main source of mismatch between client and replica. (We will also show that our MaxMind results agree with known-location RIPE Atlas results, when addressing the bias in probe locations.) Of course, the geographically closest replica may not be the lowest latency replica due to limited peering between ISPs and constrained BGP policy. In §4 and §5, when using client-sourced traceroute data from RIPE Atlas probes, we will quantify how often replica selection can be improved, not just for geographic proximity, but for reducing latency as well.

Figure 1a shows what fraction of queries and clients are directed to anycast sites ordered by rank. Only about 1/3rd of queries go to the geographically closest (rank zero) site. 31.6% of all queries go to sites ranked 5 or higher.

Figure 1a shows that 2/3 of all queries/clients are somehow "misdirected" by anycast. Figure 1b provides a measure of the cost of these errors, by quantifying the *extra* distance queries that are not directed to their closest site must travel. Figure 1b shows that over 1/3rd of the queries travel over 1000 km more than minimal, and over 8.0% travel more 5000 km extra.

These results, compiled over one year, and from over 102B queries and 35M IP addresses, representing over 190 countries, show that there is significant room for improving the

[4] We do not include measurements from G-root since it does not respond to "hostname.bind." queries with identifiers that distinguish replicas.

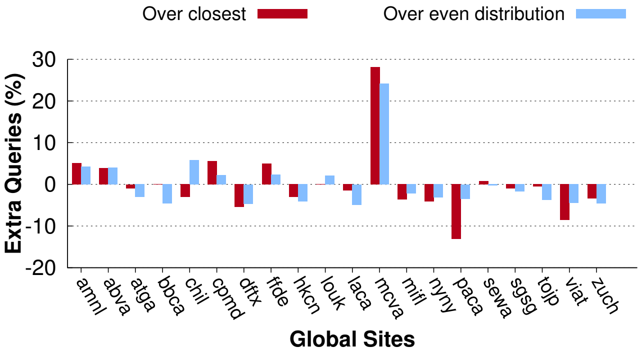
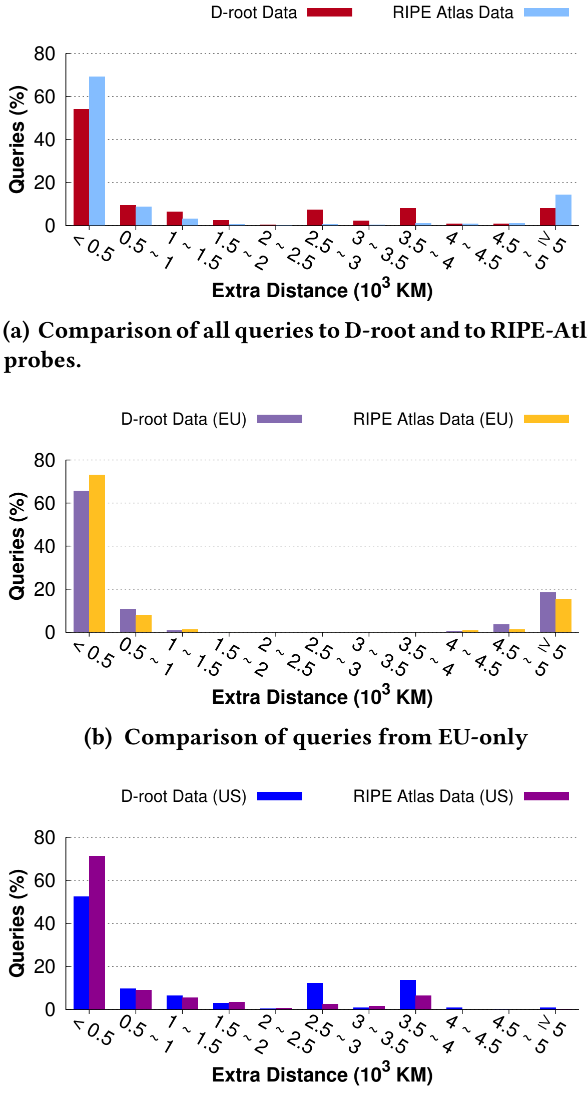


Figure 2: D-root load balance.



(a) Comparison of all queries to D-root and to RIPE-Atlas probes.

(b) Comparison of queries from EU-only

(c) Comparison of queries from US-only

Figure 3: D-root clients vs. RIPE-Atlas probes: Additional distance traveled

latency/geographic proximity behavior of Internet anycast. Next, we consider load balance: perhaps anycast's poor latency is offset by a good balance of queries to replicas?

Figure 2 shows two measures of load balance. The x-axis lists global replicas for D-root. The "Over even distribution" bars show fraction of queries, over (or under) the even distribution in which each site (ideally) receives an equal share of queries. For instance, the figure shows that the *mcva* site received 24.2% more queries than its "fair share," whereas *dftx* received 4.7% less. The "Over closest" bars show query distribution compared to the scenario when all queries were directed to their geographically closest site. We see that *mcva* received nearly 30% more queries than it would have, had all queries been directed to their closest site. By the same measure, *paca* received 13.1% fewer queries.

These results, together, show that for D-root, anycast performs poorly: it is neither effective at directing clients to nearby replicas, nor does it balance load particularly effectively. We next investigate if these trends generalize.

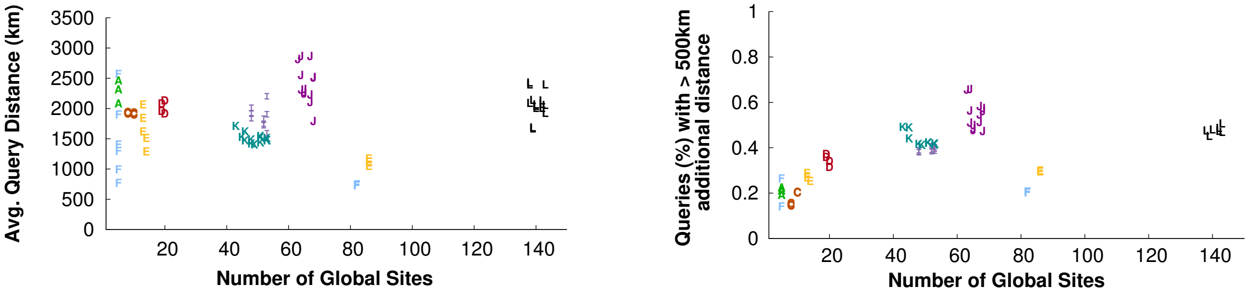
3.2 Performance across different roots

Unfortunately, we do not have access to D-root like dataset from other roots or other anycast services. Instead, we use active measurements from RIPE Atlas of D- and other roots to understand anycast behavior.

The vast majority of RIPE Atlas probes are in Europe (75%) and the United States (11%). This bias could make it appear that anycast works well (or poorly) overall if it does so only in Europe. Fortunately, at least for D-root, our trace data provides "ground truth" for how queries are distributed across sites, and we can compare RIPE Atlas results with the results compiled from the trace data.

Figure 3 plots the extra distance measure (how far beyond the geographically closest site does a query travel) for RIPE Atlas probes to D-root and compares them to D-root trace results. For the RIPE Atlas probes, we obtain their public locations[42], and then use the "hostname.bind." query to locate the D-root site the source was directed to. This figure shows data for one week for both RIPE Atlas probes and for D-root traces. Due to the bias in RIPE probe locations, we plot queries from Europe, United States, and all locations separately.

There are two main takeaways from this result. First, the RIPE probe location bias is significant, in that the results, especially outside of Europe, do not correspond particularly well with the ground truth distribution obtained at D-root. Second, in all cases, the RIPE probe results *overestimate* how well anycast performs.



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

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

Figure 4: How the number of anycast global sites affects performance. Each point represents data from a week in 2017, sampled to show at most four points per x axis value per root (for legibility). Lower y-axis values represent higher performance.

Since we have no reason to believe RIPE Atlas queries to D-root are treated differently from queries to other roots, our second conclusion leads us to believe that it is reasonable to study how poorly anycast performs using data derived from the RIPE probes. In reality, we expect anycast performance to be *worse*, as shown by the D-root data.

Figure 5 shows the extra distance measure for three roots: C, K, and L. C-root, which is operated by a Tier-1 ISP (Cogent), performs better than D-root. We expect that C performs well because replica selection is performed largely by intra-domain routing: most queries directed to C-root will be sent along an AS path that traverses "up" toward providers without geographic movement, then "across" a peering link to Cogent at the nearest location where Cogent operates (i.e., using "early-exit" routing), and once in Cogent's network, all replicas are available through intra-domain routing. There is little opportunity for a "bad" choice that, as we will see, may come from preferring one transit AS over another. Other roots, in contrast, lack a single common provider, allowing queries to be directed to ASes that can only reach a subset of replicas. K- and L-, operated by RIPE NCC and ICANN, show performance similar to D-root.

Marginal benefit of Anycast. Longitudinal RIPE Atlas measurements also allow us to understand how anycast improves performance as global sites are added.

Figure 4 plots the performance of anycast versus the number of global sites for various root servers. The x-axis is a count of global sites. For each root, we count the number of global sites in each week of 2017, and then measure its performance over that week. Therefore, there are fifty-two 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 82 sites.

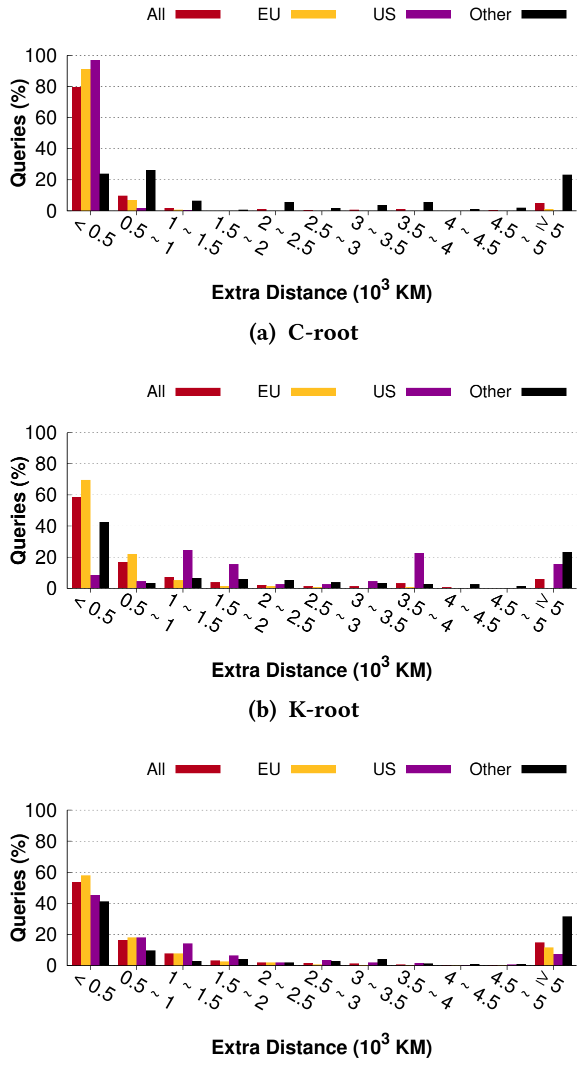
We consider two different performance measures: the left plot (a) shows the average distance traveled by RIPE Atlas queries to each root, and the right plot (b) shows the fraction of queries that had to travel more than 500 km beyond the closest site. The average distance traveled is an absolute measure of performance, and we expect this metric to decrease as the number of sites increases. The extra distance traveled is a *relative* measure of performance, since the extra distance depends on the number of available sites. Hence, the right plot measures both the performance of anycast and how efficiently new sites are utilized.

For some roots, e.g., 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 variability in routing over one year. Other roots, e.g., E-, F-, and K-root added many (77 sites for F) sites during this year, and the figure plots the effect of this investment in infrastructure. Unfortunately, even though F-root added 77 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.

These results derived from RIPE Atlas probes lead us to conclude that the performance problems shown in the D-root data are not special, but indeed representative of current anycast deployments. In the next section, we investigate whether these problems are endemic to Internet routing, or specific to anycast.

4 ANYCAST PROBLEMS

In the previous section, we have described how anycast provides neither particularly good (geographically proximate) routing properties nor balances load across sites effectively. In this section, we isolate the performance deficit into what can be attributed to routing of anycast prefixes, typically



(a) C-root

(b) K-root

(c) L-root

Figure 5: Distribution of RIPE-Atlas queries over additional distance (compared to their closest sites) traveled.

caused by choosing a poor site, and what can be attributed to unicast BGP policies and peering. Intuitively, BGP may create circuitous paths that have longer latency than the geographic distance between endpoints would require [50], and adding anycast allows the selection not only of a circuitous path, but one that does not even lead to a nearby replica. In this section, we compare these two sources of path inflation.

Suppose source s sends a query to anycast address a for a query; this query reaches site $S_{s\to a}$. Our general plan is to evaluate the performance of alternate anycast sites $S'_{s\to a}$ that *could* have been chosen for the query. Individual sites are not often directly addressable, and queries sent to the anycast address will deterministically go to $S_{s\to a}$. We devise a two step process to estimate the performance to a subset of (promising) alternate sites S':

(1) Find *unicast representatives* of each anycast site serving address a. A unicast representative for an anycast site is a unicast address u that is geographically close to the anycast site S, is contained within the AS that advertises the site, and shares (substantially) the same network path when reached from a source that is directed to that site via anycast. That is, the path from s to a shares, with s to u, the same AS path and approximate latency, when u is meant to represent the site $S_{s\to a}$.

(2) Measure the performance from source s to address a and address u to compare whether the site at u would be better than the default a.

This two step process lets us measure how well a given site *would have* performed had it been chosen by the underlying routing when queries were sent to anycast address a.

4.1 Selecting unicast representatives

Unicast IP addresses used for management of individual replicas are published for C-, K-, and L-root.[5] For these, we pick one address per site as the unicast representative address for that site. We will still evaluate below whether this management address operates as a representative, since the network could be engineered to route management traffic very differently from real queries.

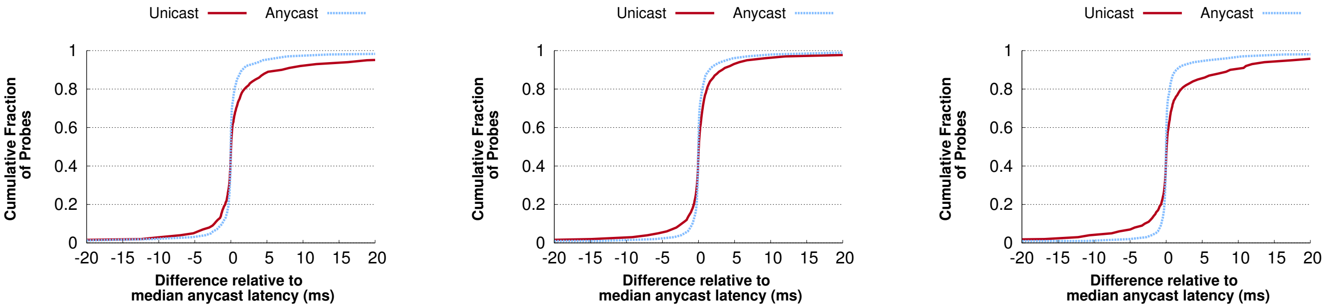
Other root DNS servers (e.g., D-root[35]) locate replicas at Internet eXchange Points (IXPs). Packet Clearing House (PCH) operates route collectors at more than 150 IXPs, and releases the BGP routing tables collected from these route collectors [36]. These routing tables provide us with other (unicast) prefixes that are reachable at the IXP, and we choose an address from the smallest unicast prefix at an IXP as the unicast representative of the colocated anycast site.[6]

4.1.1 Goodness of unicast representative. Using the method just described, we selected unicast representatives for C-, D-, K- and L-root.[7] In this section, we evaluate how well these addresses represent their anycast sites. We compare both the measured latency and the path overlap between unicast representatives and anycast sites. Recall that the RIPE Atlas probes query DNS root replicas and collect periodic traceroutes. Each of these measurements provides data about a

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

[6] 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.

[7] We omit evaluation results from L-root for space. The results for unicast representatives of L-root are similar to those of K-root. .



(a) C-root

(b) D-root

(c) K-root

Figure 6: 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.

single site per root. We augmented these probes to also measure the latency to the unicast representative of the anycast site chosen, and perform corresponding traceroutes.

The following results show the comparison of the latency and path to the anycast site via its anycast address and to the representative of the chosen anycast site via the unicast representative address. This comparison shows that the representative addresses are not routed in a way that systematically degrades (or improves) their performance. However, it is necessarily the case that the representative address is in a different prefix than the anycast address, and thus may experience different BGP-level path selection.

Figure 6 shows how latency to the unicast representative differs from the latency to the anycast address for C-, D-, and K-root. L-root, not shown, was similar to K-root.

From RIPE's built-in DNS CHAOS query measurements, we know which probe uses which site. (We confirmed that the affinity of a probe to a site is stable during measurement.) We assign probes to measure the unicast representative address corresponding to the site it used, so a different number of probes may be used to measure different sites. For each root, we aim to use about 2000 probes to measure their corresponding anycast sites and unicast representatives. We distribute those probes across sites, limiting to at most 200 probes per site for C and D, 30 probes per site for the larger K and L. Some sites will see measurements from fewer probes if too few probes use that site for anycast.

From each probe, we send traceroutes to both the anycast address and to the unicast representative of the chosen site; these will allow us to compare the AS paths. We obtain the latencies from a probe to the anycast address and to the unicast representative address. To account for ordinary variance in routing, we also obtain the median anycast latency from the probe to the anycast address during the one-hour window (leveraging RIPE's built-in ping measurements). In Figure 6, we compare the differences of our one-time measured latencies to the median anycast latency. The comparison from individual anycast measurement to median indicates a baseline (blue); the comparison between individual measurement to the unicast representative to the median anycast is a measure of representativeness (red).

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

The traceroute data from the RIPE Atlas probes allow us to evaluate the similarity in AS level paths to anycast sites versus unicast representatives. We use the method described below in §4.2 to infer AS level paths from traceroutes.

Table 1 shows a sample of sites from different roots and the fraction of the AS path that matches. Unicast representatives show a close match overall, with over 90% for C, 90% for D, 75% for K matching the 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 C-root and D-root have single hosting ASes (Cogent and PCH) from which unicast representatives are drawn, while K-root and L-root have different hosting ASes at different sites. Recall that we do not expect complete agreement, since unicast and anycast addresses are in different prefixes that may be routed differently.

4.2 AS path inference

Section 3 shows that anycast is choosing poorly. When we compare the path to the chosen anycast site to a path to the representative address of (what should be) a better site, we can determine where the two paths diverge. It is at this "decision point" that route selection failed: although there is a direct path to a representative address at a nearby site, a distant site was chosen.

We must locate the "decision point" in both geography and in the AS graph. By locating it in geography, we can infer which might be the geographically closest site to that decision point, even if it isn't the better site reached. By locating it in the AS graph, we can infer which of the two next-hop autonomous systems was not selected, which could be due to explicit policy or simple tie-breaking.

The first step in recognizing the decision point is to infer AS-level paths from IP-level paths obtained from RIPE Atlas traceroutes. Direct use of BGP routing tables, as applied in CAIDA's prefix-to-AS mapping [9], is challenging because of missing hops and multiple-origin conflicts. Here we describe how convert the traceroute path into an AS path suitable for comparison with other paths.

Mao et al. [29] proposed a heuristic method to improve IP-to-AS mapping. They collected traceroute and BGP tables from the same set of vantage points, and identified processing steps over the sequence of IP addresses necessary to construct a match to the reference path from BGP. Their approach would allow those without access to the BGP data at a particular location (e.g., at RIPE Atlas probes) to infer the BGP path associated with a readily-measurable traceroute path. Particularly, we adopted the following four steps:

- If an unresponsive/unresolved IP hop from traceroutes is between of two hops that map to the same AS, we 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. We identify such hops and remove them from inferred AS path. Nomikos and Dimitropoulos provide a tool [34] to collect IP prefixes assigned to IXPs. They collect data from PeeringDB and PCH, including prefixes for over 1000 IXPs. Using this dataset should yield better detection accuracy than the algorithm for IXP detection used in [29].
- Detect multiple origin ASes (MOAS). Once found a MOAS hop, we map it to a set of ASes. For the rest of the paper, we include these traceroutes in our comparison with other traceroutes. We 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.

Mao et al.[29] evaluated their IP-to-AS mapping algorithm. Only about 72% of traceroutes matched the corresponding BGP AS paths with basic IP-to-AS mapping using BGP tables. By applying these four steps to resolve the unmapped IP hops and IXP addresses, the matching rate increased to over 80%. Therefore, we expect that applying these processing steps will match the AS path with 80% accuracy, and that, in turn, this overall measure of agreement is a lower bound on the accuracy of suffixes of the path (after the decision point).

We do not consider traceroutes that cannot be completely resolved: if an unresponsive or unresolved IP hop lies between two different ASes, we abandon the comparison to other paths in the group we analyze below; 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 root, described in more detail below in §4.4.

4.3 Anycast and unicast path inflation

Unicast routing is subject to path inflation in which the path taken is longer than necessary. Spring et al. [50] 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. Measurements of AS path inference to unicast representatives allow us to understand if anycast is subject to *additional* path inflation.

Consider the scenario shown in Figure 7, which is derived from a real example in our dataset. Figure 7 shows a RIPE Atlas probe outside Tokyo, Japan, trying to connect to a replica for D-root. D-root hosts a global site in Tokyo; however, there is no short route (that does not traverse the United States) from the probe IP address to the D-root replica there. In this instance, anycast routes the probe to a D-root site in Los Angeles, CA. However, there *is* a unicast route from the probe to a site in Singapore, and that site is *closer* than Los Angeles in both latency and distance. In this example, the extra distance from Tokyo to Singapore can be considered unicast path inflation. (It is difficult to believe that no (perhaps policy-violating) path exists between the source and Tokyo-based replica.) However, the latency difference between probe–Singapore versus probe–Los Angeles is due *anycast path inflation.* Anycast path inflation quantifies the extra cost incurred by anycast by not choosing paths that are available via unicast.

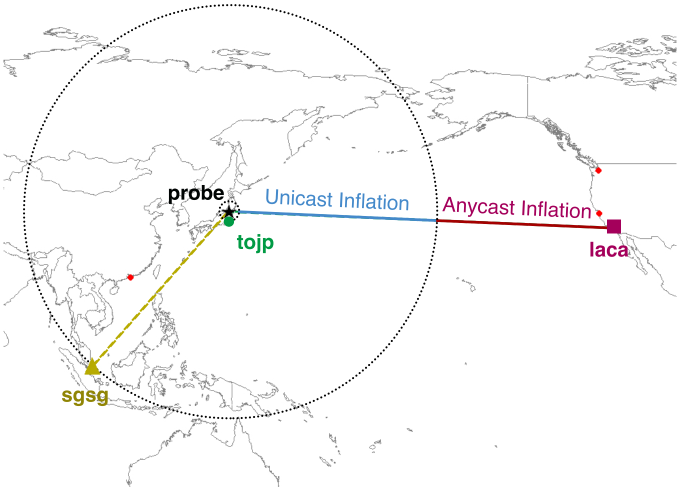


Figure 7: Illustration of anycast path inflation compared to unicast path inflation using a real example. The probe in Japan has no direct route to the closest site 'tojp' and was directed to 'laca', however 'sgsg' is the site that provides lower latency to the probe.

4.4 Quantifying anycast path inflation

The task in this section is to quantify how much of the lost performance in anycast is due to typical unicast path inflation, and how much is anycast despite the existence of a unicast path. We will err on the side of (potentially) underestimating anycast path inflation by sampling candidate representatives rather than performing an exhaustive measurement from sources to all possible alternate sites or even to all reasonably close sites.

We first need to determine the latencies to $C_{s\to a}$, the chosen anycast site, to $G_{s\to a}$, the geographically closest anycast site, and to $L_{s\to a}$, the site reachable with the lowest latency from s. The first is already obtained by RIPE in the "built-in" measurement. The second, G, is trivial to determine by tracing to the unicast representative of the nearest site to the RIPE Atlas probe.

The third, L, is more challenging because exhaustive probing is not feasible. RIPE Atlas probes are a shared resource that rate limit measurements and should be used carefully. The value of additional measurements seemed low: the amount of anycast path inflation we will see is substantial without exhaustively seeking optimal.

We focus on probes that choose an anycast site C further than 500 km beyond the closest, by geography, site, G. That is, we focus on the queries that have apparent potential to be improved; for the other queries, they experience little unicast and anycast path inflation. For C-root, we collected traceroutes from 1,862 probes that had such potential, and 1,541 of them have all complete traceroutes; for D-root, we collected traceroutes from 3,570 probes and 2,785 gave us complete traceroutes; for K-root, we collected traceroutes from 2,886 probes and 1,398 of them were complete.

We interpret the measurements as follows. If the measured RTT to the geographically closest site, $G_{s\to a}$, is less than that predicted by distance (using the Htrae constant [1], 0.0269 msec/mile) to the second closest site G', assume $L = G$. This chooses the geographically closest as the lowest-latency replica if the second closest is unlikely to be any better.

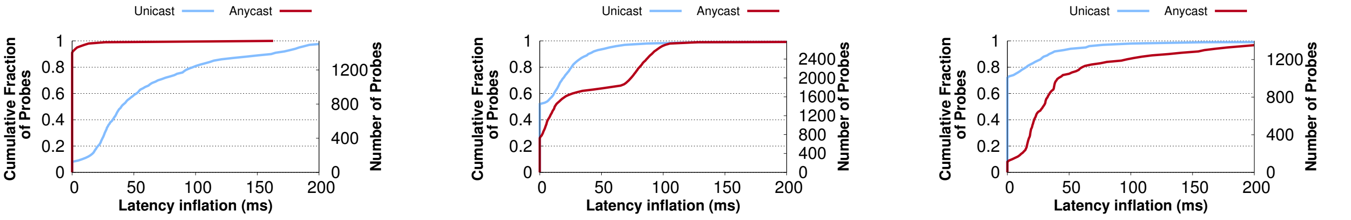
If C is already the second closest replica G', assume L is either C or G, whichever is less. Otherwise, we will measure the latency to the second closest replica and set L to the least of C, G or G'. In some cases, we may choose to include a third-closest popular replica that still is within a distance that could yield a reduction in latency.

With the latencies to C, G and L, we compute anycast path inflations and compare to unicast path inflations. Anycast path inflation quantifies the extra latency or distance when anycast does not choose the best unicast path. We compute anycast path inflation as the difference in round trip time between C and L, where round trip time to C is at least as large as the round trip time to the site with the lowest latency L. The available paths to different replicas (or representaties) have already been filtered based on BGP routing policies and thus experience unicast path inflations. Typical, unicast path inflation from BGP is captured by the difference between the round trip time to L and the predicted round trip time, by distance, to G.

Figure 8 presents unicast and anycast path inflation for 1,541 probes for C-, 2,785 for D-, and 1,398 for K-roots. For D- and K-root, anycast is unable to use the better unicast paths that are available, possibly due to route selection policy at ISPs. This is a counter-intuitive result, because it shows that extra choices provided by adding anycast sites can *decrease* performance, since ISPs may (and do) choose the "wrong" advertisement out of many available, thereby increasing the latency to the anycast prefix!

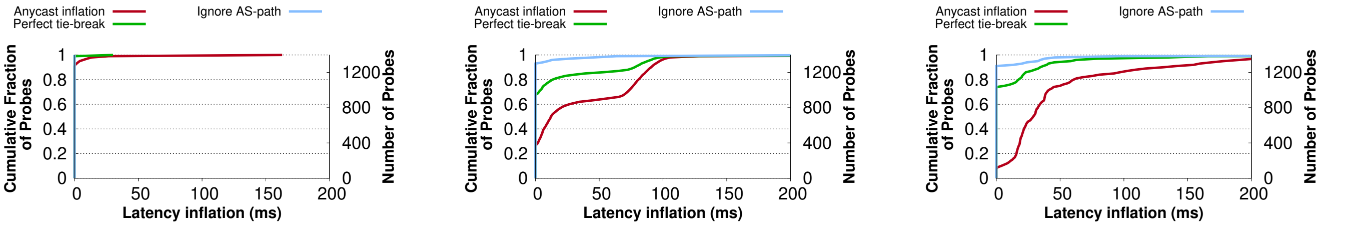
5 POTENTIAL

The previous section shows that anycast routing performs worse than unicast. ASes do not have sufficient information to make good selections. Indeed, this hints at an anomaly: adding replicas can sometimes make anycast routes *worse* as ASes pick "worse among equals." All is not lost, however. In this section, we show that relatively modest additions to BGP advertisements that encode static information about replicas would be sufficient to regain much of the lost performance. BGP has shown itself to be extensible and can be made to support this additional information; we prefer a protocol



(a) C-root **(b) D-root** **(c) K-root**

Figure 8: Comparison between unicast and anycast path inflation.



(a) C-root **(b) D-root** **(c) K-root**

Figure 9: Decomposition of anycast path inflation

based solution to one that requires connecting exclusively to a single large provider.

Derived from the same dataset as in Figure 8, Table 2 lists the number of probes that were directed to C-, D- and K-root, the number that were "correctly" routed to lowest latency sites ("Good"), and what route selection policies cause them not to. We consider common route selection policies that routers usually follow: "Valley-Free" [8], "Prefer-Customer," implying also a preference for peers over providers, and "Prefer Shortest AS-path." More detailed route selection policies are not public for most ISPs. We only consider coarse business relationship (customer, peer, provider) between ASes inferred from CAIDA's AS relationship dataset [8]. While it generally makes sense for some ISPs to apply more detailed route selection policies to maximize their profits, we believe that such routing policies are usually applied to routes with similar performance (e.g., latency, traffic load, etc.). As shown in Figure 9, routes with similar preference based on common routing policies can have very different performance in anycast. All the probes in the last column ("Unknown Tie-break") *could* have been routed to a better site without violating the common route selection policies.

			Prefer	Shortest	Unknown
Roots	Total	Good	Customer	AS-Path	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 2: Why probes do not choose closest sites.

Figure 9 shows how much of the anycast path inflation can be recovered if decision points select routes more intelligently.[9] The figure shows results for C-, D- and K-root: the anycast path inflation (red) lines correspond to anycast path inflation (same as Figure 8 and as defined in §4.4). The "Perfect tie-break" (green) lines correspond to the anycast path inflation that remains when ASes pick the route to the best site but still follow the common route selection policies. The "Ignore AS-path" (blue) lines show anycast path inflation when ASes pick the route to the best site regardless of the length of the AS-path in the received BGP advertisements.

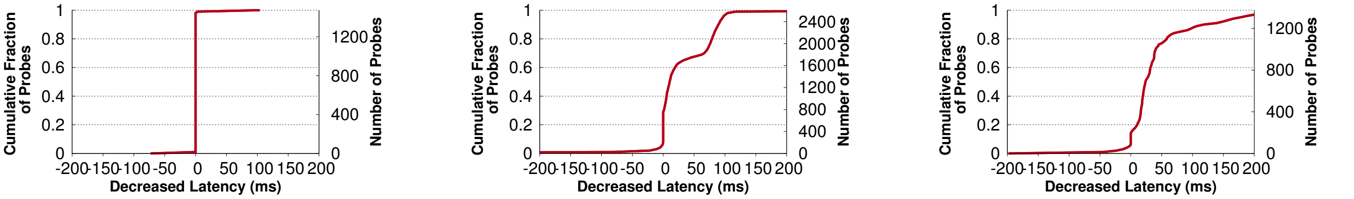
Figure 9 and Table 2 are extremely encouraging results: they show that much of the lost performance can be recovered if ASes select routes more intelligently without violating common route selection policies. Measurement-based optimization services that select the lowest latency route could be applied to anycast addresses; although such services exist for multi-homed ASes to use when choosing providers (e.g., Internap Managed Internet Route Optimizer [21].), we do not assume that their use is (or will be) sufficiently widespread in the middle of the network to improve anycast.

5.1 Static BGP Hints

Absent explicit measurement-based path selection, even a static "hint" added to BGP advertisements can prove highly

[8] We assume that the routes are compliant with "Valley-Free" since they are extracted from traceroutes.

[9] This analysis assumes that a change of route at the decision point will not change the next hop selection decisions of the ASes that receive the updated route: either they are also updated to prefer the better route or the new route with the same next-hop is no less preferable than the old.



(a) C-root **(b) D-root** **(c) K-root**

Figure 10: Geo-hints benefits for various roots.

beneficial. Consider an extension to BGP in which advertisements for anycast prefixes include the geographic location of site(s) that are reachable. When selecting routes, ASes can choose the routes to geographically closest site for each anycast prefix. Such an extension can be incrementally deployed, adds minimal overhead to advertisements, and is computationally inexpensive to evaluate when picking routes.

Each BGP router would receive advertisements for one or more sites. Higher precedence rules may cull some advertisements (e.g., an advertisement from a provider AS will be discarded in favor of advertisements from peers). Among the remaining, the router will choose the route r that advertises the geographically closest (remaining) site. If multiple do, then the router may choose arbitrarily, perhaps by which advertisement is received first. The router would then include this route r in its advertisements to BGP neighbors, as per usual. All traffic destined to the anycast prefix would be forwarded using route r.

Including explicit information about the approximate locations of reachable sites generalizes Ballani's recommendation [4] in which the anycast operator must cause remote clients to reach a provider serving all replicas by using only one upstream provider. Here, we intend to permit ASes to choose the path that reaches a nearby replica, without dynamic measurement and without requiring that the anycast operator choose a single large provider.

We evaluated our scheme through simulation over the traceroutes collected in §4.4 to the chosen, geo-closest and lowest-latency sites. Recall the "decision point" discussion in §4.2 in which the key task is to find the point of divergence between the paths to different anycast sites. We consider which sites would be listed in the BGP advertisements propagated along the routes of the traceroutes, and simulate the selection of routes that include the closest of the anycast sites to the decision point (not necessarily the closest to the source). We use "undns" [51] to track where traceroutes traversed, and thus infer which sites (among the measured ones) will be advertised to the decision point. Consider the example in Figure 7; the decision point between the traceroutes to *laca* and *tojp* is located at Los Angeles. According to the geo-hint, the route selected at the decision point should be the one that leads to *laca*. We then computed the latency difference between the geo-hinted site and the chosen site. This evaluation may underestimate the potential benefit because additional traceroutes could add new decision points that could expose a route to an even closer site.

Figure 10 shows the performance improvement that traffic destined to C-, D-, and K-root would receive using the static geographic list. The static hint does no harm to the performance of C-root, which is near ideal. Anycast to D- and K-root both show dramatic improvement. For D-root, about 1/3 of the probes improve latency by 50 msec; for K-root, 23% do. D-root shows a "step" behavior because it deploys about 20 global replicas, and for many replicas, the geo-hint is able to avoid very long latency (cross-continental/cross-oceanic) links. K-root has more than 50 global replicas, and the improvements are more evenly distributed.

Note that choosing the route that includes the closest replica site may not lead to actually using that replica. For example, should a Florida site be advertised to an AS in South America and be chosen as the path having the geographically closest site, the lowest-latency replica may not in fact be the one in Florida if paths traverse, say, Texas or Virginia along the way. In this way, the geographic list, at least as we have evaluated it with a single decision point, may choose the G replica from §4.4 over the L replica.

A simple, concrete implementation of this approach would designate community tags wherein the first 16 bits are distinct, e.g., 0xfffe to avoid conflict with the reserved 0xffff and the convention of using the first 16 to represent the AS number originating the tag, and the last 16 bits encode coarse latitude and longitude. Latitude varies -90 to 90, but inhabited latitude is more -50 to 74 [41] and can thus be encoded in 7 bits. Longitude varies -180 to 180, so can be encoded in the remaining 9 bits easily. Anycast sites would include the community tag in outgoing advertisements, these tags would propagate as 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. To implement the proposed fix, BGP routers would need to be configured with their latitude and longitude, perform computation to decode the encoded latitude and longitude in BGP community tags, and to obtain distances to the destinations in the routes. However, the lat/lon configuration is usually one-time for each router, and the computation has little overhead and only need to be executed when new routes are received. Overall, we believe the proposed fix introduces little overhead in FIB computation and should not significantly impact control plane performance. Encoding geographical information in BGP community to improve BGP selection or diagnosis is already a practical method used in many ISPs and IXPs [17, 48]. These examples suggest the practicality of the proposed fix and a means to approximate it with explicit preferences for specific sites.

We also implemented an experimental deployment on the PEERING testbed [46] to estimate whether tags propagate well enough to be used by distant ISPs. PEERING allows researchers to announce prefixes allocated to the testbed with customized BGP community tags from muxes at seven different locations. For each PEERING mux, which represents an anycast replica, we embed a specific code in the BGP community tags it announces. We announced the prefix 184.164.249.0/24 (ASN47065) from seven different locations including Amsterdam, Athens, Los Angeles, Boston, Phoenix, Seattle, Belo Horizonte (Brazil).

We characterized the propagation of community tags by collecting BGP routes towards the prefix announced from PEERING from 20 RouteViews [32] route collectors. By default, Cisco routers [10] do not pass BGP community tags to their peers. Among the 20 collectors, 11 of them received routes with customized community tags. 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 replica, i.e., they are presented with the routes to their closest replica. The other 5 received tags from their second closest replicas; another one is provided with tags to the fourth closest replica. 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 our customized community tags from PEERING testbed are treated differently from BGP communities that are already used in practice, we also characterize the propagation of community tags from other ISPs, including ServerCentral [48], Packet Clearing House [37] and Init7 [22]. We found 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. Incomplete propagation of community tags deserves dedicated study to understand how and why these they are filtered: in theory, these are transitive attributes that could help optimize routes, but their potential is limited in practice.

Other forms of hints. If BGP were to be extended to add tags specific to anycast prefixes, other forms of hints, both static and dynamic, can easily be added. One static hint would simply report only the number of sites reachable via a route. From this number, the BGP router could choose the feasible route that advertises the most sites, in the hope that one of the many will be good. This integer hint would have even lower overhead than the geographic list we have evaluated, but may miss replica sites served by smaller ISPs. It is, however, another instance of preferring the path that leads to the largest provider for an anycast address, generalizing Ballani's single-provider approach [4].

On the other end of the spectrum, measurement services could update hints based on load or latency, allowing anycast to natively approximate more sophisticated server selection algorithms that rely on extensive measurement infrastructures. A major advantage of our proposal is that regardless of hint type, it remains incrementally deployable, compatible with existing BGP policy, and should for some reason the hints be removed from advertisement (e.g., because the performance monitoring service experiences a temporary failure), performance defaults to regular BGP-based anycast behavior. Finally, the architecture is flexible enough to permit different types of hints to be added by different anycast services, and for ASes to employ their own mechanisms to evaluate hints and choose the best route.

6 CONCLUSION

IP anycast serves as the foundation of some of the most critical network infrastructure, and yet its inefficiencies have long gone misunderstood and unfixed. Using passive and active measurements, we have presented an in-depth root-cause analysis of the inefficiencies of root DNS servers' IP anycast deployments. Our results empirically validate an earlier hypothesis [4] that equal-length AS paths are largely to blame for anycast latency inflation. Guided by these findings, we presented a fix that reduces anycast inflation through the use of geo-hints: small geographic hints included in BGP to help routers more efficiently choose from among multiple equal-length AS paths. Unlike prior proposals [3, 4], geo-hints are easily and incrementally deployable. Crucially, geo-hints demonstrates that IP anycast can be efficient without having to rely on the cooperation of a single large upstream provider.

ACKNOWLEDGMENTS

We thank the anonymous reviewers, and our shepherd, Olivier Bonaventure, for their helpful comments on the paper. This work was supported in part by NSF awards CNS-1409249, CNS-1526635 and CNS-1619048.

REFERENCES

[1] S. Agarwal and J. R. Lorch. Matchmaking for online games and other latency-sensitive P2P systems. In *ACM SIGCOMM*, 2009.

[2] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe. A practical architecture for an anycast CDN. In *ACM Transactions on the Web (TWEB)*, volume 5, 2011.

[3] H. Ballani and P. Francis. Towards a global IP anycast service. In *ACM SIGCOMM*, 2005.

[4] H. Ballani, P. Francis, and S. Ratnasamy. A measurement-based deployment proposal for IP anycast. In *ACM Internet Measurement Conference (IMC)*, 2006.

[5] N. Brownlee, K. C. Claffy, and E. Nemeth. DNS root/gTLD performance measurements. In *USENIX LISA*, 2001.

[6] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the expansion of Google's serving infrastructure. In *ACM Internet Measurement Conference (IMC)*, pages 313–326, 2013.

[7] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the performance of an anycast CDN. In *ACM Internet Measurement Conference (IMC)*, pages 531–537, 2015.

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

[9] Center for Applied Internet Data Analysis (CAIDA). Routeviews prefix to AS mappings dataset for IPv4 and IPv6. http://www.caida.org/data/routing/routeviews-prefix2as.xml.

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

[11] Cloudflare, Inc. Announcing 1.1.1.1: the fastest, privacy-first consumer DNS service. https://blog.cloudflare.com/announcing-1111/.

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

[13] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky. Evaluating the effects of anycast on DNS root name servers. In *RIPE document RIPE-393*, 2006.

[14] X. Fan, E. Katz-Bassett, and J. Heidemann. Assessing affinity between users and CDN sites. In *International Workshop on Traffic Monitoring and Analysis (TMA)*, pages 95–110. Springer, 2015.

[15] A. Flavel, P. Mani, D. A. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev. Fastroute: A scalable load-aware anycast routing architecture for modern CDNs. In *Symposium on Networked Systems Design and Implementation (NSDI)*, 2015.

[16] D. Giordano, D. Cicalese, A. Finamore, M. Mellia, M. Munafò, D. Z. Joumblatt, and D. Rossi. A first characterization of anycast traffic from passive traces. In *International Workshop on Traffic Monitoring and Analysis (TMA)*. Springer, 2016.

[17] V. Giotsas, C. Dietzel, G. Smaragdakis, A. Feldmann, A. Berger, and E. Aben. Detecting peering infrastructure outages in the wild. In *ACM SIGCOMM*, pages 446–459, 2017.

[18] B. Gueye, S. Uhlig, and S. Fdida. Investigating the imprecision of IP block-based geolocation. In *Passive and Active Network Measurement Conference (PAM)*, pages 237–240. Springer, 2007.

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

[20] B. Huffaker, M. Fomenkov, and kc claffy. Geocompare: A comparison of public and commercial geolocation databases. Technical report, Center for Applied Internet Data Analysis (CAIDA), 2011.

[21] INAP Inc. InterNAP Managed Internet Route Optimizer. http://www.inap.com/network-services/miro-controller/, 2017.

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

[23] D. Katabi and J. Wroclawski. A framework for scalable global IP-anycast (GIA). In *ACM SIGCOMM*, 2000.

[24] J. H. Kuipers. Analyzing the K-root DNS anycast infrastructure. In *Twente Student Conference on IT*, 2015.

[25] B.-S. Lee, Y. S. Tan, Y. Sekiya, A. Narishige, and S. Date. Availability and effectiveness of root DNS servers: A long term study. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 862–865, 2010.

[26] M. Lentz, D. Levin, J. Castonguay, N. Spring, and B. Bhattacharjee. D-mystifying the D-root address change. In *ACM Internet Measurement Conference (IMC)*, 2013.

[27] J. Liang, J. Jiang, H. Duan, K. Li, and J. Wu. Measuring query latency of top level DNS servers. In *Passive and Active Network Measurement Conference (PAM)*, pages 145–154. Springer, 2013.

[28] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and kc claffy. Two days in the life of the DNS anycast root servers. In *Passive and Active Network Measurement Conference (PAM)*, pages 125–134, 2007.

[29] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *ACM SIGCOMM*, pages 365–378, 2003.

[30] MaxMind Inc. MaxMind GeoIP2 city. https://www.maxmind.com/en/geoip2-databases, 2017.

[31] C. Metz. IP anycast point-to-(any) point communication. volume 6, pages 94–98. IEEE, 2002.

[32] D. Meyer. University of Oregon Route Views project. http://www.routeviews.org/.

[33] G. Moura, R. d. O. Schmidt, J. Heidemann, W. B. de Vries, M. Muller, L. Wei, and C. Hesselman. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In *ACM Internet Measurement Conference (IMC)*, pages 255–270, 2016.

[34] G. Nomikos and X. Dimitropoulos. traIXroute: Detecting IXPs in traceroute paths. In *Passive and Active Network Measurement Conference (PAM)*, pages 346–358. Springer, 2016.

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

[36] Packet Clearing House (PCH). PCH daily routing snapshots. https://www.pch.net/resources/Routing_Data/.

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

[38] J. Pang, J. Hendricks, A. Akella, R. De Prisco, B. Maggs, and S. Seshan. Availability, usage, and deployment characteristics of the Domain Name System. In *ACM Internet Measurement Conference (IMC)*, pages 1–14, 2004.

[39] C. Partridge, T. Mendez, and W. Milliken. Host anycasting service, Nov. 1993. RFC 1546.

[40] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP geolocation databases: Unreliable? In *ACM SIGCOMM Computer Communication Review (CCR)*, volume 41, pages 53–56, 2011.

[41] Radical Cartography. World's population in 2000, by latitude. http://www.radicalcartography.net/index.html?histpop, 2017.

[42] RIPE NCC. RIPE atlas probes information. https://atlas.ripe.net/probes/, 2017.

[43] RIPE NCC Staff. RIPE atlas: A global internet measurement network. In *Internet Protocol Journal*, volume 18, 2015.

[44] Root-servers.org. Root servers archives. http://root-servers.org/archives/.

[45] S. Sarat, V. Pappas, and A. Terzis. On the use of anycast in DNS. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*, pages 71–78, 2006.

[46] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, and E. Katz-Bassett. Peering: An AS for us. In *Workshop on Hot Topics in Networks (HotNets)*. ACM, 2014.

[47] R. d. O. Schmidt, J. Heidemann, and J. H. Kuipers. Anycast latency: How many sites are enough? In *Passive and Active Network Measurement*

Conference (PAM), pages 188–200. Springer, 2017.

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

[49] Y. Shavitt and N. Zilberman. A geolocation databases study. In *IEEE Journal on Selected Areas in Communications*, volume 29, pages 2044–2056, 2011.

[50] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, 2003.

[51] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with rocketfuel. In *ACM SIGCOMM Computer Communication Review (CCR)*, volume 32, pages 133–145, 2002.

[52] M. Weinberg and D. Wessels. Review and analysis of anonmalous traffic to A-root and J-root on Nov/Dec 2015. In 24th DNS-OARC Workshop, 2016.