Internet Anycast: Performance, Problems and Potential

Paper #24. 13 pages

ABSTRACT

Internet Anycast depends on interdomain 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 interdomain routing policies that can increase path length compared to theoretical optimals (e.g., geographically closest). Unlike unicast, anycast routes are also affected by poor tie-break choices 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-specific path inflation, enabling anycast to approach the performance of unicast routing.

1 INTRODUCTION

Anycast is one of the fundamental modes of communication, in which a set of anycast *instances* 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 instances offer an equivalent service [3, 15, 28]. This basic *one-to-any* form of communication has become the basis for critical network infrastructure; all root DNS servers are hosted via IP anycast [18], and some content delivery networks (CDNs) use it in an attempt to lower latencies and distribute load [7].

What makes IP anycast such an attractive option when deploying a globally replicated service is the mental model that it would appear to permit. In particular, as one adds more anycast instances in locations with many clients, it is generally believed [3, 7, 10] 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 increased global deployment of DNS root anycast instances, 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 thousands of kilometers away from their closest instance [16, 20], resulting in increased latency [4]. 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 proposed solution comes from Ballani et al. [3, 4], who hypothesize that IP anycast is efficient only when all of a service's instances are hosted by a single upstream provider. Were this the only solution, it would mean that efficiently running a georeplicated service over IP anycast would require cooperation from a large ISP; adding even a small ISP could negatively impact performance. Is this a fundamental limitation of IP 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 of IP anycast, why it fails (and succeeds), and how to fix it without relying upon a single large upstream provider. The paper makes three broad, interrelated contributions:

Performance (§3): Using passive and active measurements of several 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 (after all, interdomain routing as a whole in no way guarantees optimality), we find the inefficiencies to be surprisingly excessive. In particular, we show that adding more anycast instances (1) often *increases* overall latency, and (2) often exacerbates load balancing, matching clients to anycast instances in different continents than their own.

Problems (§4): To explore the root causes of these performance issues, we introduce a novel measurement technique that allows us to compare the AS-level paths from clients to multiple IP anycast instances. The resulting data indicates

[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.

that the majority of IP anycast inefficiency is due to BGP *tie-breaks*: routers are presented with two or more IP anycast instances each of whom have equal-hop-length paths. Lacking any useful information to distinguish between them, routers often select a distant, high-latency anycast instance over the closer, low-latency one. Again, it is not surprising that interdomain 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 that routers can use to more intelligently break ties among equal-hop-length alternatives. We find that this reduces the "anycast inflation" (the additional latency imposed by IP anycast) to *zero* for over 65% of clients. This technique is incrementally deployable and, although we evaluate it only on root DNS data, 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, we will make our tools and nonsensitive datasets publicly available.

2 RELATED WORK

IP anycast [15, 28] is a widely used technique that allows services to be transparently replicated across the Internet. Two of the most studied applications of IP anycast to date are root DNS servers [18, 23] and content delivery networks (CDNs) [2, 6, 7, 11–13].

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, 27].

Performance measurements of IP anycast Several studies have compared the RTTs between clients and their anycast instances to the *smallest* RTT among all of the possible anycast instances [4, 9, 10, 19, 33].

Early studies of the performance of IP anycast among DNS root servers indicated a promising trend towards lowered latency. In 2006, Sarat et al. [33] 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 instance, the latency overheads are typically small (75th percentiles of less than 5 msec for K-root and less than 20 msec for F-root). In 2010 (before D-, E-, and H-root had deployed anycast), Lee et al. [17] 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 instances are added—and identifies a root cause (BGP tie-breakers) and a fix.

Most recently, in 2016, Schmidt et al. [10] 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 IP anycast deployments, adding more instances *harms* performance by increasing latency—a phenomenon originally predicted by Ballani et al. [4]. Like Schmidt et al. [10], 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 instances. In 2006, Liu et al. [20] 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 [16] performed a somewhat 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 instance. Our findings largely reinforce these prior results by showing that IP anycast can indeed be surprisingly far from optimal, but we expand them by identifying the root cause of these inefficiencies (BGP tie-breakers) 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 instance, but have offered no concrete explanation or fix. 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 instances 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 tie-breaker issues 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 small 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 one or more 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. In this paper, we consider each root to be an 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?**

There are many different metrics that a server selection mechanism may optimize or improve. These include, but are not limited to, access latency, load balance, resilience, and geographic proximity. Our goal is to study whether anycast is effective at improving these metrics.[2] In particular, we consider the marginal benefit of adding replicas: how do 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 probes from RIPE Atlas probes. We describes these datasets, including their features and limitations, next.

[2]Evaluating and improving anycast resilience is not within the scope of this paper; 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 [23, 35]. 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.

Root server traffic traces. Our first source of data is sampled traffic from the sites of a specific root DNS server.[3] As of Jan 2018, X root had over 120 anycast sites, 20 of which were global and the rest local. 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, X 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, load variance, and inter-site traffic variation, each of which we analyze.

There are two limitations to the X-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 immediately clear if the performance for this one root extrapolates to anycast performance in general. Second, this data is entirely passively collected, and does not provide insight into alternate AS paths or other selection polices. To address both these problems, we augment this dataset with active measurements.

RIPE Atlas measurements. The RIPE Atlas framework [30, 32] is a set of ~10,000 probes distributed across 180 countries and in ~3587 ASes as of Jan 2018. Each probe periodically executes pre-defined measurements, called "build-in measurements", that include specific DNS queries and traceroutes to all 13 DNS roots.

Our analysis uses queries that the RIPE Atlas probes sent to the 9 out of 13 roots that have at least 5 anycast global sites. 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) is directed to by anycast over time. We augment this data with our own measurements of alternate replicas and addresses in order to evaluate possible alternatives (§4).

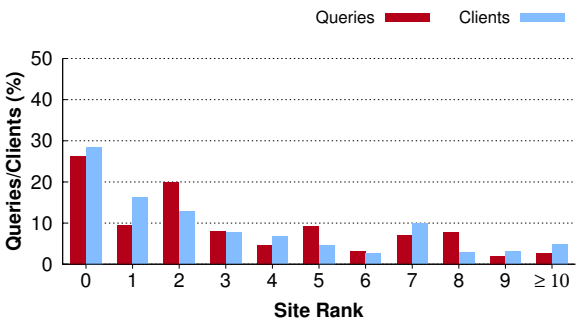
3.1 How does anycast perform?

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

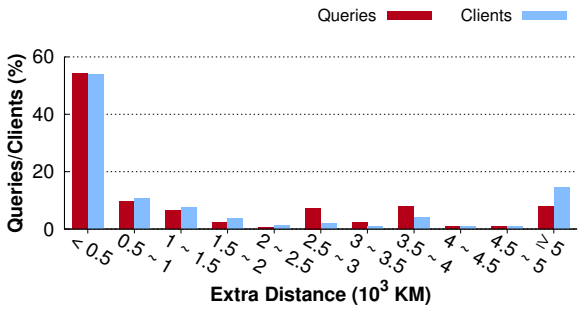
Figures 1a and 1b show a measure of goodness of anycast for X-root. For each query received at X-root, we geo-locate the source of the query by IP address using the Maxmind

[3]To preserve author anonymity, we refer to this server as X-root.

[4]We were unable to include measurements from G-root since it does not respond to "hostname.bind" DNS CHAOS queries with meaningful identifiers that we can use to distinguish replicas.



(a) X-root queries by rank



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

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

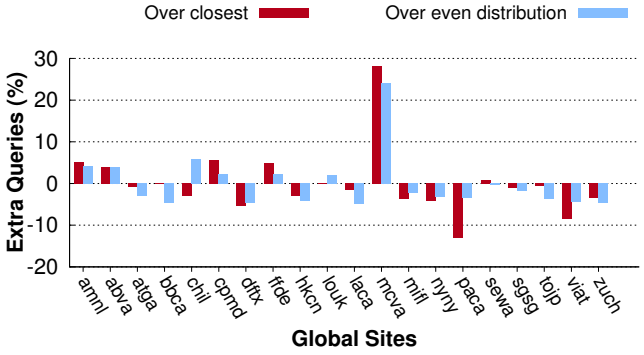


Figure 2: X-root load balance.

database [22]. Next, we measure the distance from the query source to all X-root sites. For a query, the closest site is ranked 0, the next closet rank 1, and so on. We compute the same measure for each source IP address (client) as well.

We use geographic distance as an intuitive approximation of expected latency because the passive trace dataset taken at replicas does not provide a direct measure of client latency. For various reasons, including limited peering between ISPs and constrained BGP policy, the geographically closest replica may not be the lowest latency replica, and thus a distant replica could possibly be the best. 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/clients are directed to ordered by rank. For instance, the figure shows that only about 1/3rd of queries go to the geographically closest site. 31.6 % of all queries/clients 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 1000km more than minimal, and over 8.0% travel more 5000km 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 latency/geographic proximity behavior of Internet anycast. Next, we consider load balance — perhaps, anycast's poor latency behavior is offset by providing reasonable distribution of queries to replicas?

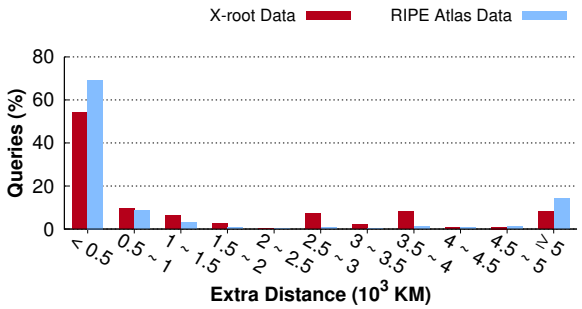
Figure 2 shows two measures of load balance. The x-axis lists global replicas for X-root. The "Over even distribution" bars shows fraction of queries, over (or under) the even distribution received by each site, where even distribution is defined as each site receiving an equal number of queries. For instance, the figure shows that the *mcva* site received 24.2% more queries than its "fair share" , whereas the *dftx* received 4.7% less. The "Over closest" bars show query distribution compared to the scenario when all queries were directed to their geographically closet 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 X-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.

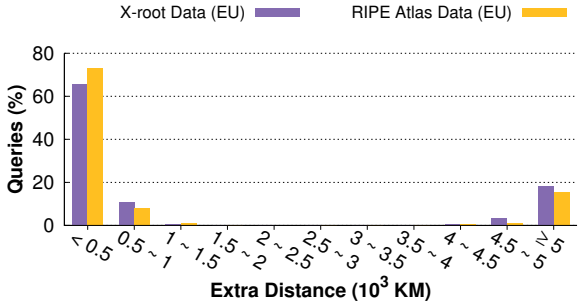
3.2 Performance across different roots

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

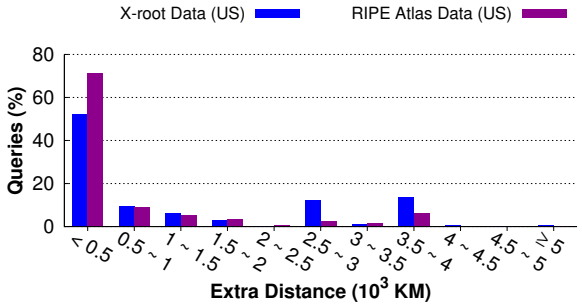
Before presenting our results, we note that the location of RIPE Atlas probes is biased, with the vast majority in



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



(b) Comparison of queries from EU-only



(c) Comparison of queries from US-only

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

Europe (75%) and the United States (11%). We must take this bias into account if we are to extrapolate results based on RIPE probes. Fortunately, at least for X-root, our trace data provides "ground truth" for how queries are distributed across sites, and we compare RIPE probe 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 X-root and compares them to X-root results. For the RIPE probes, we obtain their public locations[31], and then use the `hostname.bind` query to locate the X-root site the source was directed to. This figure shows data for one week for both RIPE Atlas probes and for X-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, we note that 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 X-root. Second, we note that in all cases, the RIPE probe results *overestimate* how well anycast performs.

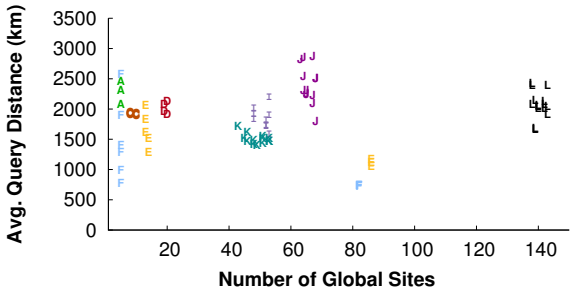
Since we do not have any specific reason to believe RIPE Atlas queries to X-root are treated any differently than queries to other roots, or other anycast, our second conclusion leads us to believe that it is reasonable to study how poorly anycast performs using data derived from the RIPE probes. In reality, we expect anycast performance to be *worse*, as shown by the X-root data.

Figure 5 shows the additional distance measure (how far beyond the geographically closest site does a query travel) for three roots: C, K, and L. C-root, which is operated by a Tier-1 ISP (Cogent), performs better than X-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 (i.e., no AS advertises the anycast address despite having access to only a subset of replicas). There is little opportunity for a "bad" choice that, as we will see, may come from preferring one AS over another. K- and L-, operated by RIPE NCC. and ICANN, respectively, show performance similar to X-root.

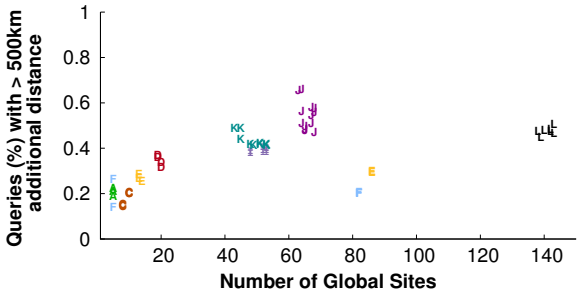
Marginal benefit of Anycast. Analysis of longitudinal RIPE Atlas measurements also allows us to understand how anycast improves performance as sites are added. (Note that a site may add replicas, but that is not considered in our analysis.)

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 500km farther than the closest site. The average distance traveled is an absolute



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



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

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.

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 X-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 study how much of the performance deficit can be attributed specifically to routing of anycast prefixes, typically involving choosing a poor site, and contrast that performance deficit with that expected of typical unicast routing caused by BGP policies and peering. Intuitively, BGP may create circuitous paths that have longer latency than the geographic distance between endpoints would require, as described in [34], 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 messages 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.

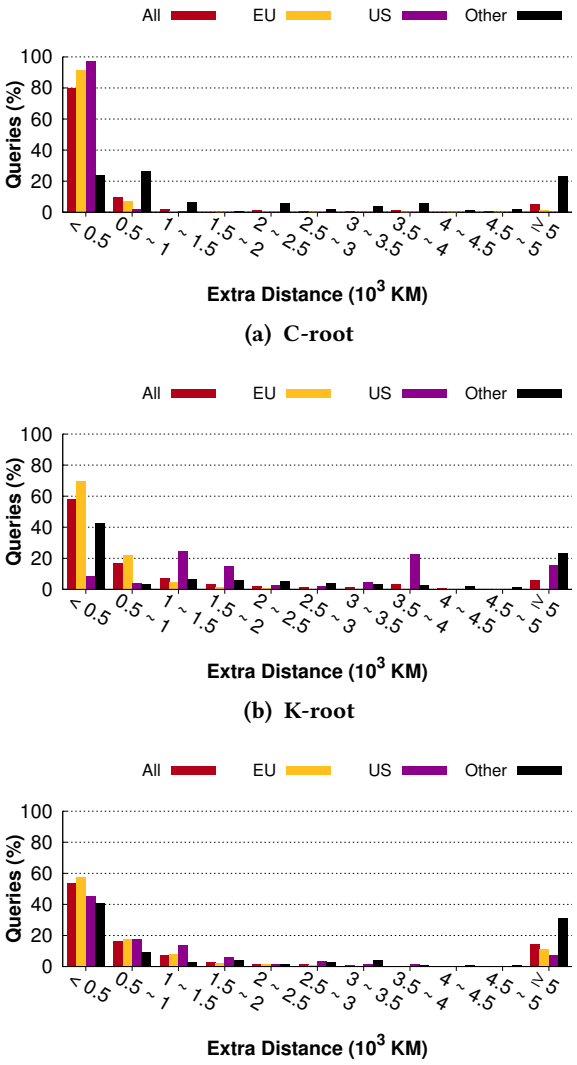


Figure 5: Distribution of RIPE-Atlas queries over additional distance (compared to their closest sites) traveled.

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.

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

Other root DNS servers (e.g., D-root[25]) 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 [26]. 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. 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 also perform periodic traceroutes. Each of these measurements provide us data about a 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.

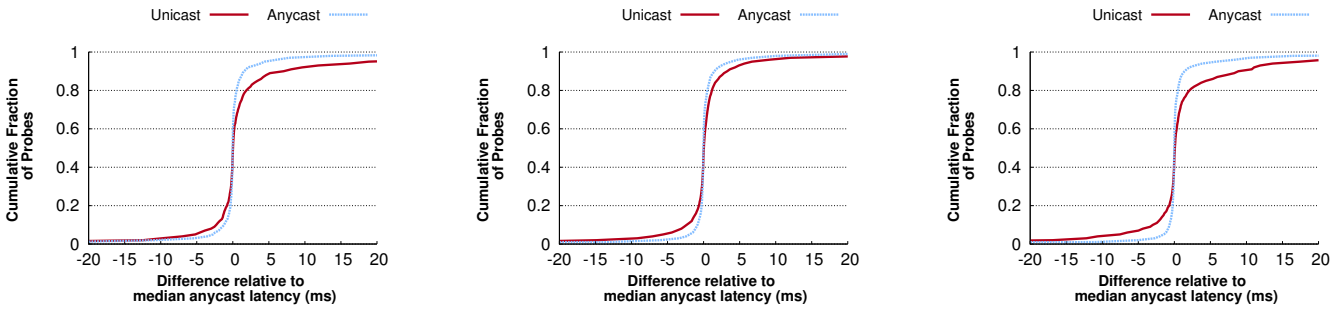
The following comparison of the performance and path to the anycast site via its anycast address and to the representative of the chosen anycast site via the unicast representative address shows that the representative addresses are not routed in a way that systematically degrades (or improves) their 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 the natural

[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.



(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.

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.

variance in routing during time, we also obtain the median anycast latency from the probe to anycast address during the one-hour window (leveraging RIPE's built-in ping measurements). Then, we compare the differences of our one-time measured latencies to the median anycast latency, and results are shown in Figure 6. The comparison from individual anycast measurement to median indicates a baseline; the comparison between individual measurement to the unicast representative to the median anycast is a measure of representativeness.

The traceroute data from the RIPE 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, and 85% for L-root 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 with a path to a representative address of what appears to be a better site, we can determine where the two paths diverge. It is at this "decision point" that route selection failed: although a good path exists to the representative address, which is in the AS serving the anycast address in a geographic location that has a replica, a path to a different site was preferred.

We must locate the "decision point" where the paths diverged, 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 [8], is challenging because of missing hops and multiple-origin conflicts. Here we describe how we approach converting the traceroute path into an AS path suitable for comparison with other paths.

Mao et al. [21] 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 we used for traceroute measurements, we cannot apply their method directly. However we 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, 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 [24] 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 [21].
- 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.

According to the evaluation in [21], with basic IP-to-AS mapping using BGP tables, only about 72% of traceroutes matched the corresponding BGP AS paths. By applying the four steps above to resolve the unmapped IP hops and IXP addresses, the matching rate increased above 80%. Based on this, we expect that applying these techniques will infer 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, described in more detail below in §4.4.

4.3 Anycast-specific path inflation

Unicast routing is subject to path inflation in which the path taken is longer than necessary. Spring et al. [34] 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 and AS path inference to unicast representatives allow us to understand if anycast is subject to *additional* path inflation.

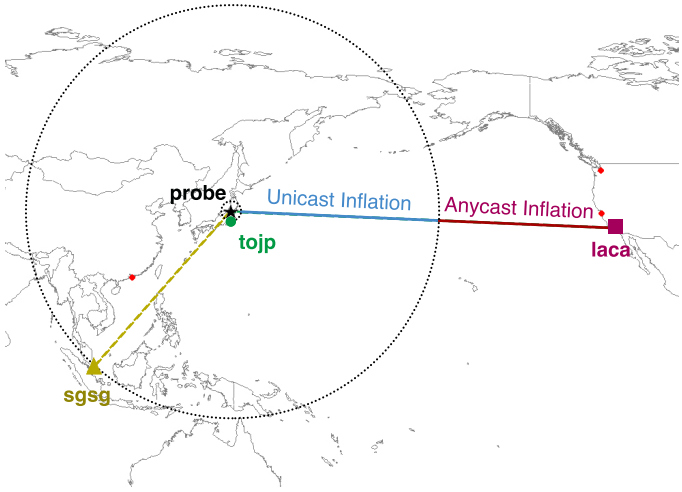


Figure 7: Illustration of anycast-specific inflation compared to unicast inflation using a real example. The probe in Japan has no direct route to the closest site 'tojp' and was directed to 'laca', however 'sgsg' is the site that provides lower latency to the probe.

Consider the scenario shown in Figure 7, which is derived from a real example in our dataset. Figure 7 shows a RIPE 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 this example, the extra distance from Tokyo to Singapore can be considered inflation due to intra-domain policy. (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 inflation.* Anycast-specific inflation quantifies the extra cost incurred by anycast by not choosing paths that are available via unicast. In this section, we quantify anycast-specific inflation, and try to understand the underlying reasons.

4.4 Measurement methodology

The task in this section is to quantify how much of the lost performance in anycast replica selection is due to typical BGP path inflation, and how much is anycast-specific despite the existence of a unicast path. We will err on the side of (potentially) underestimating anycast-specific 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 $S_{s\to a}$, 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 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-specific 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 C-root, we collected traceroutes from 1862 probes that had such potential, and 1541 of them have all complete traceroutes; for D-root, we collected traceroutes from 3570 probes and 2785 gave us complete traceroutes; for K-root, we collected traceroutes from 2886 probes and 1398 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 ms/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 evaluate "anycast" inflations and compare them with "unicast" inflations. "Anycast" inflation is 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. Typical, "unicast" inflation from BGP is captured by the difference between the round trip time to L and the predicted latency, by distance, to G.

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 2: Why probes do not choose closest sites.

4.5 Quantifying anycast-specific inflation

Using the measurement technique described above, we can attribute path inflation to unicast inflation or anycast-specific inflation. Routes from a probe incur the unicast-specific inflation because of common policies including "Prefer-Customer", implying also a preference for peers over providers, "Valley-Free", and "Prefer Shorter AS-path".

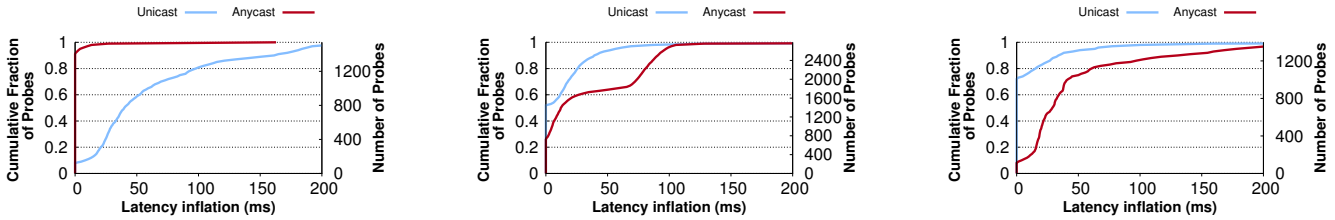
Anycast inflation quantifies the additional inflation when anycast does not choose available unicast paths. Note that the available paths have already been filtered based on the unicast path inflation, and represent cases when multiple paths are available to providers, and a long path is chosen.

Figure 8 presents unicast- and anycast-specific inflation for 1541 probes for C-, 2785 for D-, and 1398 for K-roots. The results show that for D- and K-root, even though better paths are available, anycast is unable to utilize them, possibly due to poor tie-breaking rules at ISPs. This is a counter-intuitive result, because it shows that extra choices provided by adding 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

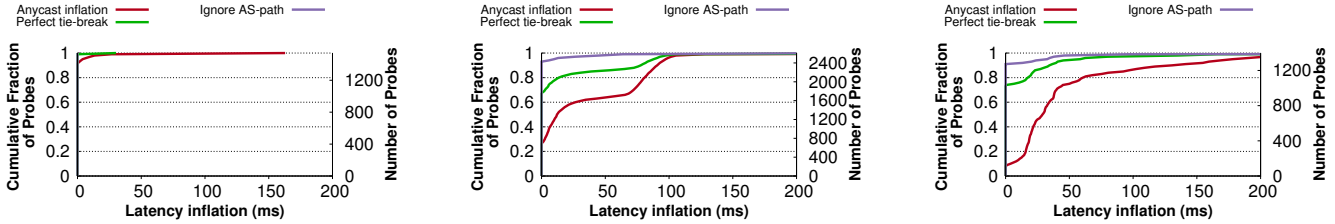
The previous section shows that anycast routing performs worse than unicast. ASes do not have sufficient information to make good selections. Indeed, this hints at an anomaly: adding replicas can sometimes make anycast routes *worse* as ASes pick "worse among equals". All is not lost, however. In this 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 based solution to one that requires connecting exclusively to a single large provider.

Figure 9 shows how much of the anycast-specific inflation can be recovered if decision points tie-break more intelligently. The figure shows results for C-, D- and K-root: the anycast inflation (red) lines correspond to the inflation due to anycast (same as Figure 8 and as defined in §4.4). The "Perfect tie-break" (green) lines correspond to the anycast inflation that remains when ASes pick the "best" unicast



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

Figure 8: Comparison between unicast inflation and anycast-specific inflation.



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

Figure 9: Decomposition of anycast-specific inflation

site but still prefer shorter AS-paths. The "Ignore AS-path" (purple) lines show anycast inflation when ASes pick the best unicast site regardless of the length of the AS-path in the received BGP advertisements. For this dataset, 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, and the reasons why the others were not. All the probes in the last column ("Tie-break") *could* have been routed to a better site without violating BGP rules.

Figure 9 and Table 2 are extremely encouraging results: they shows that much of the lost performance can be recovered if ASes tie-break more intelligently. 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 [14].), we do not assume that their use is (or will be) sufficiently widespread in the middle of the network to improve anycast.

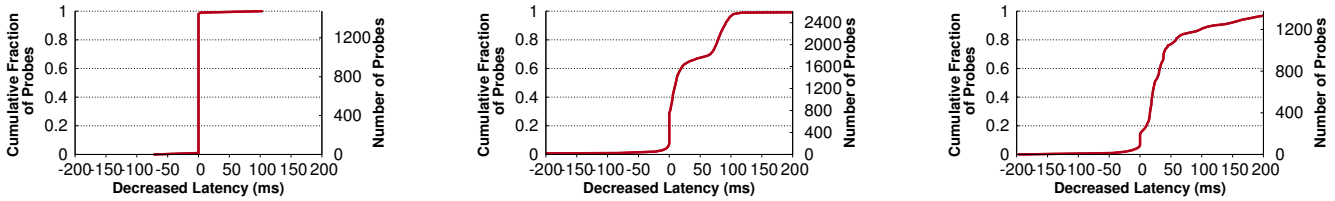
5.1 Static BGP Hints

Absent explicit measurement-based path selection, even a static "hint" added to BGP advertisements can prove highly beneficial. Consider an extension to BGP in which advertisements for anycast prefixes include the geographic location of site(s) that are reachable. When tie-breaking, ASes can choose the 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 one or more advertisements, each advertising 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 anycasted prefix would be forwarded using route r.

Including explicit information about the approximate locations of reachable sites generalizes the recommendation in [4] in which the anycast operator must compel remote clients to reach a provider serving all replicas by using only one 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 have evaluated such a scheme. Recall the "decision point" discussion in §4.2 in which the key task is to identify the point of divergence between the path to the chosen anycast site and the lower-latency anycast site. We consider which sites would be listed in advertisements from both options (the chosen and the better) and simulate the selection of the advertisement that includes the nearest of the replica sites to the decision point (not necessarily the closest to the source). Note that our evaluation may underestimate the potential benefit of hints as widespread deployment could



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

Figure 10: Geo-hints benefits for various roots.

add new decision points that could expose a path to an even closer replica.

Figure 10 shows the improvement traffic destined to C-, D-, and K-root would receive using the static geographic list. Note that the static hint does no harm: performance of C root, which is near ideal, is not adversely affected. Anycast 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. 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 nearest replica site may not lead to actually using that nearest replica. For example, should a Florida site be advertised to an AS in South America and be chosen as the path having the geographically nearest 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 [29] 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.

Other forms of hints. If BGP were to be extended to add attributes specific to anycast prefixes, other forms of hints, both static and dynamic, can easily be added. One static hint is to simply report 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 also 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.

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. *ACM Transactions on the Web (TWEB)*, 5(4):17, 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. Claffy, and E. Nemeth. DNS Root/gTLD performance measurements. *USENIX LISA, San Diego, CA*, 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. ACM, 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. ACM, 2015.

[8] 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.

[9] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky. Evaluating the effects of anycast on DNS Root name servers. *RIPE document RIPE-393*, 6, 2006.

[10] R. de Oliveira 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.

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

[12] 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. *connections*, 27:19, 2015.

[13] 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. IFIP, 2016.

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

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

[16] J. H. Kuipers. Analyzing the K-root DNS anycast infrastructure. 2015.

[17] 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 *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 862–865. IEEE, 2010.

[18] 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.

[19] 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.

[20] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, et al. Two days in the life of the DNS anycast root servers. In *Passive and Active Network Measurement Conference (PAM)*.

[21] Z. M. Mao, J. Rexford, J. Wang, and R. 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.

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

[23] 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 *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 255–270. ACM, 2016.

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

[25] Packet Clearing House (PCH). D-Root Peering Policy. https://www.pch.net/services/dns_anycast.

[26] Packet Clearing House (PCH). PCH Daily Routing Snapshots. https://www.pch.net/resources/Routing_Data/.

[27] 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. ACM, 2004.

[28] C. Partridge, T. Mendez, and W. Milliken. *Host Anycasting Service*, Nov. 1993. RFC 1546.

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

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

[31] RIPE NCC. RIPE Atlas Probes. https://atlas.ripe.net/probes/, 2017.

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

[33] S. Sarat, V. Pappas, and A. 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.

[34] N. Spring, R. Mahajan, and T. Anderson. Quantifying the Causes of Path Inflation. In *ACM SIGCOMM*, 2003.

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