Anycast Latency: How Many Sites Are Enough?

Ricardo de O. Schmidt[1], John Heidemann[2], and Jan Harm Kuipers[1]

[1]University of Twente
[2]USC/Information Sciences Institute
r.schmidt@utwente.nl,johnh@isi.edu,j.h.kuipers@student.utwente.nl

Abstract. Anycast is widely used today to provide important services such as DNS and Content Delivery Networks (CDNs). An anycast service uses multiple *sites* to provide high availability, capacity and redundancy. BGP routing associates users to sites, defining the *catchment* that each site serves. Although prior work has studied how users associate with anycast services informally, in this paper we examine the key question *how many anycast sites are needed* to provide good latency, and the worst case latencies that specific deployments see. To answer this question, we first define the *optimal performance* that is possible, then explore how routing, specific anycast policies, and site location affect performance. We develop a new method capable of determining optimal performance and use it to study four real-world anycast services operated by different organizations: C-, F-, K-, and L-Root, each part of the Root DNS service. We measure their performance from more than 7,900 vantage points (VPs) worldwide using RIPE Atlas. (Given the VPs uneven geographic distribution, we evaluate and control for potential bias.) Our key results show that a few sites can provide performance nearly as good as many, and that geographic location and good connectivity have a far stronger effect on latency than having many sites. We show how often users see the closest anycast site, and how strongly routing policy affects site selection.

1 Introduction

Internet content providers want to provide their customers with good service, guaranteeing high reliability and fast performance. These goals can be limited by underlying resources at servers (load) and in the network (throughput, latency, and reliability). Replicating instances of the service at different *sites* around the Internet can improve all of these factors by increasing the number of available servers, moving them closer to the users, and diversifying the network in between.

Service replication is widely used for naming (DNS) and web and media Content Delivery Networks (CDNs). Two different mechanisms associate users with particular service instances: DNS-based redirection [12] and IP anycast [1,30], and they can be combined [13,28]). When the service is DNS, IP anycast is the primary mechanism, used by many operators, including most root servers, top-level domains, many large companies, and public resolvers [22,38]. IP anycast is also used by several web CDNs (Bing, CloudFlare, Edgecast), while others use DNS-based redirection (Akamai, Google, and Microsoft), or their combination (LinkedIn). This paper, however, focuses only on IP anycast.

In IP anycast, service is provided on a specific *service IP address*, and that address is announced from many physical locations (*anycast sites*), each with one or multiple servers[1]. BGP routing policies then associate each user with one site, defining that site's *catchment.* Optimally users are associated with the nearest site, minimizing latency. BGP provides considerable robustness, adapting to changes in service or network availability, and allowing for some policy control. However, user-to-site mapping is determined by BGP routing, a distributed computation based on input of many network operators policies. Although mapping generally follows geography [27], studies of routing have shown that actual network topology can vary [36], and recent observations have shown that the mapping can be unexpectedly chaotic [6,23].

Anycast has been widely studied, typically with measurement studies that assess anycast coverage and latency [8,21,9,25,29,17,5,34,26], and also to enumerate anycast sites [19]. Latency studies using server-side traces show that anycast behaves roughly as expected—many geographically distributed sites reduce latency. These studies also show surprising complexity in how users are assigned to anycast sites. While prior studies cover what *does* happen, no prior work defines what *could* and *should* happen—that is, what latency is *possible*, and the reasons actual latency may differ from this ideal.

The **main contribution** of this paper is to develop a *new measurement methodology that identifies optimal latency* in IP anycast systems (§ 2), enabling a first evaluation of how close actual latencies are to their potential. Our insight is that we can determine optimal anycast latency by measuring unicast latency to *all* anycast sites of a system, providing a comparison to the assigned site by BGP. Thus, while prior work reports only latency for the selected anycast site, we can see when catchments differ from optimal and then study why. Our dataset from this study is publicly available at `http://traces.simpleweb.org/`.

Our **second contribution** is to carry out a *measurement study of four IP anycast deployments:* the C-, F-, K- and L-Root DNS services, consisting of more than 240 sites together. These services have different architectures and deployment strategies, that we study from around 7,900 RIPE Atlas probes worldwide, creating a rich dataset to inform our understanding of anycast latency.

The **final contribution** of this work is what we learn from this first comparison of actual and optimal anycast latency. Our central question is: *How many anycast sites are "enough" to get "good" latency?* To answer this question, we must first answer several related questions: Does anycast give good absolute performance (§ 3.1)? Do users get the closest anycast site (§ 3.2)? How much does the location of each anycast site affect the latency it provides overall (§ 3.3)? How much do local routing policies affect performance (§ 3.5)? With these questions resolved, we return to our key contribution and show that a modest number of well-placed anycast sites—**as few as twelve**—can provide nearly as good performance as many (§ 3.6). We also show that more sites improve the tail of the performance distribution (§ 3.4).

---

[1] The term anycast *instance* can refer to a site or to specific servers at a site. Because of this ambiguity we avoid that term in this paper.

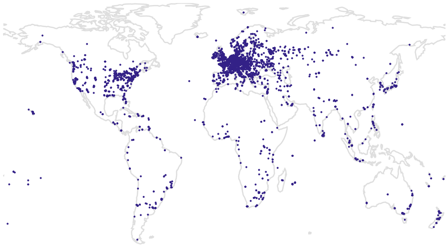


Fig. 1. Locations of more than 7,900 vantage points we use from RIPE Atlas.

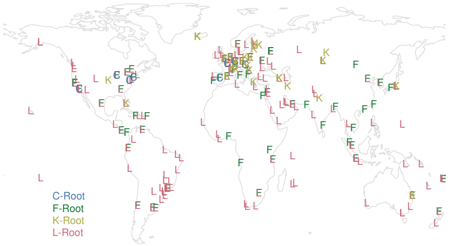


Fig. 2. Locations of sites for each service (each site is identified by its letter).

This paper focuses on anycast *latency*. We consider latency because it motivates huge investments, such as Google's 2013 expansion to thousands of locations [12], gradual expansion of Root DNS anycast to more than 500 sites [18], and CDN design in multiple companies. We recognize that anycast serves other purposes as well, including distributing load, improving resilience to Denial-of-Service attacks, and to support policy choices. These are, however, out of the scope of this paper. Our population of vantage points is European-centric (§ 3.3); while this skew affects our specific results, it does not change our qualitative conclusions. Broader exploration of CDNs, other metrics, and other sets of vantage points are future work (some in-progress).

2 Measurement Methodology

Our approach to observe anycast latency is straightforward: from as many locations (*vantage points*, or VPs) as we can, we measure *latency to all anycast sites* of each *service* that we study. These measurements approximate the *catchment* of VPs that each site serves. We use RIPE Atlas probes as VPs, and we study the C-, F-, K- and L-Root DNS services as our targets. We measure latency with pings (ICMP echo requests), and identify sites with DNS CHAOS queries. Prior studies [19,6] have used both of these mechanisms, but only to preferred site; to our knowledge, we are the first to measure latency to *all anycast sites* from all VPs, the key that allows us to study optimal latency (not just actual), and to explore policy questions (§ 3).

Measurement sources: We use more than 7,900 VPs (probes) in the RIPE Atlas framework [32]. Figure 1 shows the locations of all VPs: these cover 174 countries and 2927 ASes. We maximize coverage by using all probes that are available at each measurement time. The exact number, shown in Table 1, varies slightly as VPs come and go over measurements taken in 2015 and 2016. While RIPE VPs are global, their geographic distribution does not exactly match that of the overall Internet population. We show in § 3 that this skew strongly affects the specific *quantitative* latencies we observe, favoring sites and VPs in Europe. But it *does not* affect our *qualitative* results about the number of anycast sites and the effects of routing policies.

Measurement targets: We study four operational anycast services: the C-, F-, K- and L-Root DNS services [18] (Figure 2). Each service is run by a different

Table 1. Summary of each root service, its size in sites, and their routing policy; measurement date and number of VPs then available; how many hits are optimal, latency for each type of hit, and the cost of mishits (§ 3.2). We measure K-Root both before (K) and after (NK) its change in routing policy (§ 3.5).

letter	**sites**	**date**	**VPs**	**hit type**		**median RTT** (ms)				**mishit penalty** (ms)		
	(local)			optimal	*mishit*	all	optimal	*mishit*	*(pref.)*	25%ile	50%ile	75%ile
C	8 (0)	2015-09	5766	84%	*16%*	32	28	*61*	*55*	*2*	*5*	*10*
F	58 (53)	2015-12	6280	44%	*56%*	25	12	*39*	*20*	*8*	*15*	*51*
K	33 (14)	2015-11	6464	*41%*	*59%*	32	14	*43*	*23*	*8*	*18*	*42*
NK	36 (1)	2016-04	5557	*40%*	*60%*	30	12	*41*	*19*	*9*	*18*	*48*
L	144 (0)	2015-12	5351	24%	*76%*	30	11	*47*	*16*	*10*	*24*	*82*

operator and is optimized to meet their goals. They are diverse in both number of sites (with C small, F and K mid-sized, and L numerous), and in routing policy: all C and L sites are global (available to all), while many K and most F sites are local (service limited to specific AS). To identify optimal possible latency (§ 3), we chose these services because they all make public the *unicast* IP address of each site. We measure K Root both in 2015 (K), and again in 2016 (NK—*New K*) after major changes on its anycast policies, discussing implications in § 3.5.

Measuring anycast catchments: We map the catchments of each anycast service by observing DNS CHAOS queries [39] (with name `hostname.bind` and type `TXT`) from each VP. The reply to each VP's CHAOS query indicates its anycast site, as determined by BGP routing. The exact contents of the reply are service-specific, but several root operators (including C, F, K and L) reply with the unicast hostname of the reached site. For example, a reply for C Root is `lax1b.c.root-servers.org`, where `lax` gives the geographic location of the replying site and `1b` identifies the replying server within the site. The resolution of this name gives the unicast IP address of that server. Sites sometimes have multiple servers, but we treat all servers at a site as equivalent.

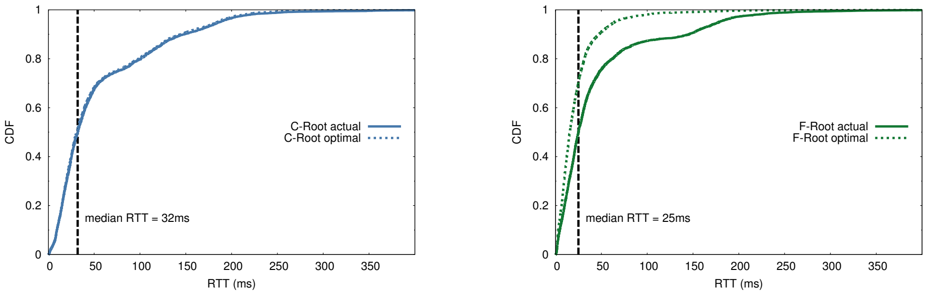
Measuring latency: We use ICMP ECHO requests (pings) to measure *latency* from VPs to both the public anycast service address (BGP-assigned site), and the unicast address of all sites for each service. To suppress noise in individual pings, we use multiple pings and report the 10th-percentile value as the measured latency. On average VPs send 30 pings to each anycast site, but the exact number varies due to dynamics on the RIPE Atlas framework, limitations on availability of probes, and measurement scheduling.

3 Observation and Findings

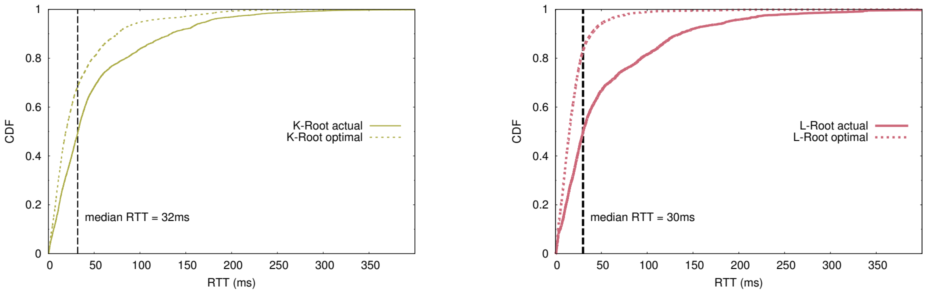
3.1 Does Anycast Give Good Absolute Performance?

We first look at absolute latency seen from VPs for each anycast service. The solid lines in Figure 3 show the distribution of latency seen from each VP to the service of the four measured letters. It reports the *actual* RTT to each VP's BGP-assigned site. We see that *all letters provide low latency to most users*: median RTT for C and K Root is 32 ms, L's median is 30 ms and F's is 25 ms.

Is 30 ms latency "good"? For DNS during web browsing (DNS on `www.example.com`), every millisecond matters. However, names at the root (like `com`)



(a) C-Root (b) F-Root



(c) K-Root (d) L-Root

Fig. 3. Distribution of RTT to all four measured letters: *optimal* RTT ignoring BGP assignment (dotted line) compared to all *actual* RTT (solid line).

are easily cachable: there are only around 1000 names and they allow caching for two days, so shared caches at recursive resolvers are *very* effective. But we consider 30 ms great, and somewhat arbitrarily define 100 ms as high latency (matching ideal network latencies from New York to California or Sydney). More study is needed to understand the relationship between Root DNS performance and user-perceived latency to provide definitive thresholds.

This data shows that *median latency does not strictly follow anycast size*—while F and L have better latency than C and K, corresponding with their larger number of anycast sites (58 and 144 vs. 8 and 33), the improvement is somewhat modest. Actual latency is no more than 30 ms different between any letter in most of the distribution. (At the tail of the distribution however, this difference increases up to 135 ms.) This result is quite surprising since there is a huge difference on the sizes of the anycast deployments of the measured letters. For services with many sites, *careful route engineering can also make a large difference in latency*. F's median latency is lower than L's (25 ms vs. 30 ms), even though it has about half the sites (58 vs. 144). This difference may be from route engineering by F, explicitly using RIPE Atlas for debugging [6].

3.2 Do Users Get the Closest Anycast Site?

While we showed a few sites can provide good latency, do they provide *optimal* latency? Anycast relies on BGP to map users to sites, but BGP only approximates shortest-path routing. The dotted lines in Figure 3 show the *optimal possible* performance based on unicast routing to each individual site of all measured letters, ignoring anycast routing policies and catchments. We see that C-Root's actual service is very close to optimal (solid and dotted lines nearly overlap). We believe that this is because C has only a few, geographically distributed sites, and all sites are global—that is, C's sites are all visible across the Internet.

By contrast, larger anycast deployments show a larger difference between actual and optimal latency. These differences arise because more sub-optimal choices are available, and because these services have some or many local nodes that might place policy limitations on routing (§ 3.5). Looking at optimal possible performance in Figure 3 we see that routing freedom would improve median latency for F-, K- and L-Root by 16 ms, 19 ms and 14 ms, which represents an improvement of 36%, 40% and 53% respectively. (We recognize that constraints on routing may be a condition of site deployment, but we wish to understand the *potential* optimal absent such constraints.)

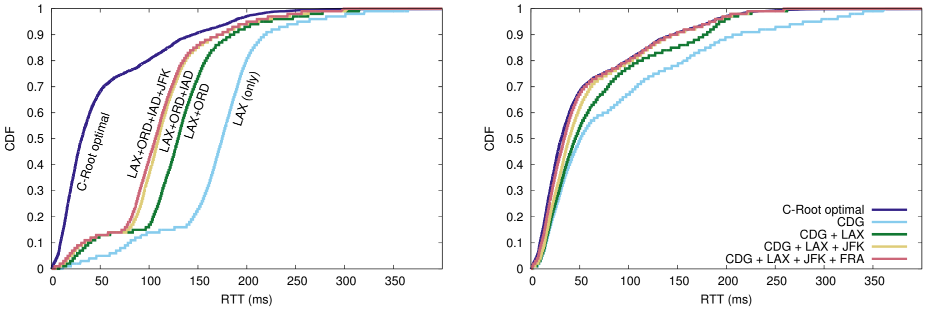
We define *mishits* as the cases when VPs are sent to sites other than the lowest latency. Table 1 shows how often mishits occur for each measured letter. Missing the nearest site often has a serious cost: the median RTT for VPs that mishit is 40 ms or higher for all letters. These large latencies are reflected in large penalties: the difference between latency cost of the mishit relative to the best possible choice (*i.e.,* optimal hit ignoring BGP). Table 1 shows the 25, 50 and 75th percentiles of the distribution of mishit penalties to all four letters.

Surprisingly, C-Root's few sites also have the lowest penalty of mishitting (median of 5 ms). We believe that this low penalty is because C's site are well connected and relatively close to each other (in the U.S. or Europe), so missing the closest often results in finding another site on the same continent, incurring little additional delay. In fact, 70% of all mishits for C-Root reached a site in the same continent as their optimal hit. The opposite is seen for L-Root, which shows the highest mishit penalty (median of 24 ms). L's many sites give many opportunities for mishit, and mishits incur much greater latency, often being served by a distant site with a global routing policy. (Consequences of mishits and differences in the distribution tail are discussed in § 3.4.)

3.3 Effects of Anycast Location on Latency and Observation Bias

It is well known that *no single location can provide equally low latency to the global Internet*, one motivation for the use of anycast by root letters. We next show that the latency of anycast service is affected more by site *location* than the absolute *number* of sites, and consider how to manage bias due to the location of our VPs. For this study we draw locations from C Root to simulate artificial services of different sizes. We then estimate client latency assuming all VPs choose their closest site (an optimistic assumption, but close, as shown in 3.2).

Effects of Site Location: Figure 4a compares the RTT distribution of four subsets of C-Root's U.S.-based sites to C-Root's optimal. The subsets begin on



(a) U.S.-based sites

(b) U.S. and Europe-based sites

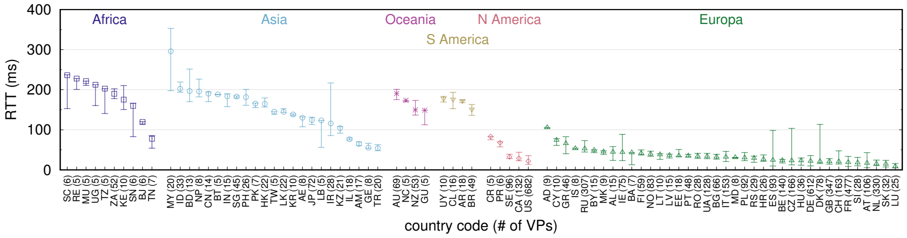
Fig. 4. Distribution of RTT to two different anycast services with 1 to 4 sites.

the right using a single location in Los Angeles (LAX), then sites are added going eastward until New York (JFK). As each site is added, the distribution shifts to the left, improving performance. In all configurations, 80% of VPs see relatively large latencies: from 150 ms for LAX-only down to 75 ms for the four-site configuration. This trend reflects speed-of-light from European VPs to the U.S., with latency improving as sites closer to Europe are added.

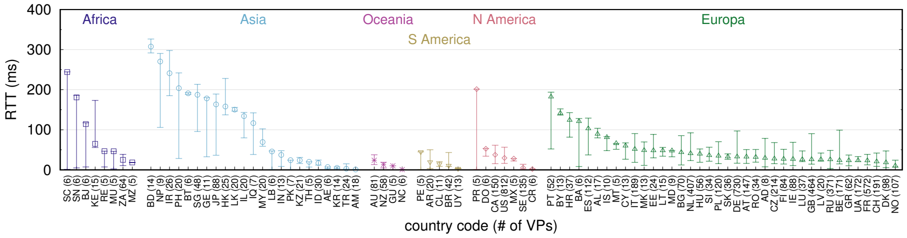
Effects of VP Location: The analysis in Figure 4a shows our measurements are dominated by the many RIPE VPs in Europe (Figure 1), characterizing a bias that weights our quantitative results to services with sites in Europe. However, this *bias in VP location does not change our qualitative conclusion that site location dominates latency.* In addition, this bias is reflected in measurement tools based on RIPE Atlas, such as DNSMON [31], and others have recognized that RIPE Atlas does not represent all global traffic [33].

Low latency with geographically distributed locations: While Figure 4a shows a pessimal selection of locations, we can minimize latency by selecting geographically distant sites. Figure 4b again compares the RTT distribution of four subsets of C-Root's sites, but now mixing sites located in U.S. and in Europe. We start with a site in Paris (CDG), close to the majority of our VPs in Europe, and with a tail elsewhere in the world—this configuration is within 20% of optimal (as defined by C's 8 sites). We then add U.S. west (LAX) and east (JFK) coasts, and then Frankfurt (FRA), each pulling the distribution closer to optimal, particularly in the tail. With the four-site combination, we virtually reach C's optimal possible performance. This data shows that *geographically distributed anycast sites can improve latency for the most distant users.* Wide geographic distribution helps because mature networks become well-connected, with latency converging down to the speed-of-light (in fiber) limit. Although both network topology and routing policies mean network and geographic proximity may diverge [36], dispersion in geography correlates with network dispersion.

Finally, comparing these figures shows that *site location matters more than number of sites.* Four ideally positioned sites do well (the CDG, LAX, JFK, and



(a) C-Root



(b) L-Root

Fig. 5. Median RTT (quartiles as error bars) for countries with at least 5 VPs (number of VPs per country is given between parenthesis). Letters at top indicate continents.

FRA line in Figure 4b is leftmost), while four poorly chose sites are far from optimal (compare the LAX, ORD, IAD, JFK line against optimal in Figure 4a).

3.4 How Much Do "Many Sites" Help?

A key result of Figure 3 is that the four letters provide roughly similar latency across most VPs, in spite of an 18× more sites (C- and L-Root show similar median latencies, 32 ms vs. 30 ms). While many sites does not affect median latency, *more sites help the tail of the distribution*, from 70th to 90th percentiles. To evaluate this tail, we next examine each country with at least 5 VPs. (We omit countries with fewer to avoid potential bias from bad "last miles" [3].)

With countries grouped by continent, Figure 5 reports the median latency for C- (Figure 5a) and L-Root (Figure 5b). Latency is highest for countries in Africa and Asia for both roots, and also in Oceania and South America for C-Root. We expect high latency for C-Root in these areas because its anycast sites are only in Europe and North America. With global anycast sites, high latency for L-Root is surprising. Using our 100 ms threshold for high latency (§ 3.1), we observe that C has about 38 countries above that threshold, while L has only about 21. L's many additional sites improve latency, but not everywhere. Somewhat more troubling is that L shows high latency for several European countries (Portugal, PT; Belarus, BY; Croatia, HR; Bosnia, BA; and Spain, ES). Even with European sites, routing policies send traffic from these countries to long distances.

When we look at countries with highest latency in Figure 5, L's many sites do improve *some* VPs in each country, as shown by the lower quartiles. However, the high median shows that these improvements are not even across all VPs in these countries. This wide variation suggests interconnection inside these countries can be poor, resulting in good performance for those VPs in ISPs that have a local anycast site, while VPs in other ISPs have to travel long distances. For example, from all 20 VPs in the Philippines (PH), 7 VPs are able to reach their optimal L sites located in the Philippines itself, with average RTT of 18 ms. The other 13 VPs, however, reach L sites in U.S. and Australia, seeing average RTT of 56 ms. None of the "unlucky" 13 VPs are within the same ASes than the other 7 "lucky" ones. We therefore conclude that *routing policies can drastically reduce the benefits of many sites.*

3.5 Do Local Anycast Policies Hurt Performance?

Anycast sites are often deployed with a *local* routing policy, where the site is only available to the hosting AS, or perhaps also directly adjacent ASes. An important question in anycast deployments is how much these policies impact on performance. The anycast deployments we studied allow us to answer if policy routing matters. The similar distributions of latency among the four letters we study (Figure 3) show that policy does not matter much. C- and L-Root place no restriction on routing, while about half of F- and most of K-Root sites are local in our initial study (Table 1). We also observe K after they changed almost all sites to global (NK in Table 1).

We study *mishits* to get a more detailed look into this question. In Table 1, mishits are VPs that do not hit the optimal site. We have examined mishits based on those that go to local or global sites in detail in our technical report [35]. Due to space, we summarize those findings here and refer that report for the detailed analysis. We see that a fair number of VPs are prevented from accessing their nearest site because they instead go to a global site: this case accounts for about 58% of F-root VPs that mishit, and 42% of K-Root mishits. Thus, restrictive local routing does add latency; and relaxing this policy could improve median latency from 37 ms to 19 ms in F-Root, and from 43 ms to 25 ms in K-root.

K-Root provided a natural experiment to evaluate if relaxing routing helps. After our initial measurements of K-Root in 2015-11, K changed all but one site to global routing; our NK dataset re-examines K-Root in 2016-04, after this policy change. Comparing K and NK in Table 1, we see only modest changes in latency: 2 ms drop in median latency, and no real change in the fraction of mishits. From discussion with the K-Root operators, we learned that local routing policies were inconsistently applied (routing limits were often ignored by peers), thus routing policies can be dominated by routing bugs.

Our main conclusion is that *careful examination and debugging routing polices of local sites* can make a large difference in performance. Bellis' tuning of F-Root anycast routing showed that debugging can improve performance [6].

3.6 How Many Sites?

Given this analysis, how many sites are needed for reasonable latency? § 3.1 shows minimal difference for median latency from 8 to 144 sites, suggesting 8 sites are reasonable based on C-Root measurements from RIPE Atlas. If we consider two sites per six continents for some redundancy, and account for under-representation of VPs in some areas, we suggest **twelve sites can provide**

reasonable latency. We caution that this number is only a rough suggestion—by no means do we suggest that 12 is perfect but 11 or 13 is horrible. This count considers only latency; we recognize more sites may be needed for many other reasons (for example, DDoS-defense and many dimensions of diversity), and it applies to an individual IP anycast service, not DNS or a CDN, which often employ multiple, independent IP anycast services. It assumes geographic distribution (§ 3.3) and that routing problems allow use of geographically close sites (§ 3.4 and § 3.5), and effective DNS caching (§ 3.1).

4 Related Work

The DNS Root has been extensively studied in the past. CAIDA's measurement infrastructure `skitter` [11] has enabled several early studies on DNS performance [8,9,21,25]. In 2004, Pang *et al.* [29] combined probing and log analysis to show that only few DNS servers were being used by a large fraction of users. Following works studied the performance of DNS, focusing on latency between clients and servers [5,17,34]. DNS CHAOS has been used to study client-server affinity [7,34]. Liu *et al.* [27] used clients geolocation to estimate RTT, and others evaluated the effect of route changes on the anycast service [4,10]. Liang *et al.* [26] used open resolvers to measure RTT from the DNS Root and major gTLDs. Bellis [6] carried out a comprehensive assessment of latency in F Root's anycast, fixing faulty route announcements to improve performance. Other work [14,24] used large and long-term datasets to show that the expansion of the anycast infrastructure improved overall performance of the Root DNS. Finally, Calder *et al.* [13] studied the choice of anycast or LDNS for redirection to CDN services.

Our work differs from these prior studies in methodology and analysis. We build on prior studies, but define *optimal* possible performance and measure it with probes to unicast addresses of all sites. This new methodology allows our analysis to go beyond measurements of what happens to statements about what *could* happen, allowing the first answers about effects of routing policy. In addition, this methodology allows us to estimate performance of alternate anycast infrastructures that are subsets of current deployments, enabling strong conclusions about the effect of numbers of sites on latency.

Furthermore, complementing our work are studies that enumerate and characterize content delivery services that use IP anycast. To exemplify some, Calder *et al.* [12] used EDNS client subnet (ECS) and latency measurements to characterize Google's serving infrastructure. Streibelt *et al.* [37] also used ECS to study Google's, Edgecast's and CacheFly's ancyast user to server mapping. Fan *et al.* [19] combined DNS queries and traceroutes to study the anycast at TLDs. Cicalese *et al.* [16] used latency measurements to geolocate anycast services, and later characterize IPv4 anycast adoption [15]. Fan *et al.* [20] combined ECS and open resolvers to measure Google's and Akamai's front-ends. Finally, Akhtar *et al.* [2] proposed a statistical approach for comparing CDNs performance.

5 Conclusions

We studied four real-world anycast deployments (the C-, F-, K- and L-Root DNS nameservers) with 7,900 VPs (RIPE Atlas probes) to systematically explore the relationship between IP anycast and latency. Unique to our collection is the combination of latency to each VP's current site, and to *all* sites, allowing evaluation of optimal possible latency. We collected new data for each of the measured services in 2015 and revisited K-Root in 2016 to evaluate changes in its routing policies. Our methodology opens up future directions, including assessment of anycast for resilience to Denial-of-Service and load balancing in addition to latency reduction.

Our new ability to compare actual to optimal latency allows us untangle several aspects of our central question: *how many anycast sites are "enough"*. Our data shows similar median performance (about 30 ms) from 8 to 144 sites, suggesting that **as few as twelve sites can provide reasonable latency**, provided they are geographically distributed, have good local interconnectivity, and DNS caching is effective.

Acknowledgments: We thank Geoff Huston (APNIC), George Michaelson (APNIC), Ray Bellis (ISC), Cristian Hesselman (SIDN Labs), Benno Overeinder (NLnet Labs) and Jaap Akkerhuis (NLnet Labs), Duane Wessels (Verisign), Paul Vixie (Farsight), Romeo Zwart (RIPE NCC), Anand Buddhdev (RIPE NCC), and operators from C Root for their technical feedback.

This research uses measurements from RIPE Atlas, operated by RIPE NCC.

Ricardo Schmidt's work is in the context of SAND (Self-managing Anycast Networks for the DNS: http://www.sand-project.nl) and DAS (DNS Anycast Security: http://www.das-project.nl) projects, sponsored by SIDN, NLnet Labs and SURFnet.

John Heidemann's work is partially sponsored by the U.S. Dept. of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via SPAWAR Systems Center Pacific under Contract No. N66001-13-C-3001, and via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement numbers FA8750-12-2-0344 and FA8750-15-2-0224. The U.S. Government is authorized to make reprints for Governmental purposes notwithstanding any copyright. The views contained herein are those of the authors and do not necessarily represent those of DHS or the U.S. Government.

References

1. J. Abley and K. E. Lindqvist. Operation of Anycast Services. RFC 4786, 2006.
2. Z. Akhtar, A. Hussain, E. Katz-Bassett, and R. Govindan. DBit: Assessing Statistically Significant Differences in CDN Performance. In *IFIP TMA*, 2016.
3. V. Bajpai, S. J. Eravuchira, and J. Schönwälder. Lessons Learned From Using the RIPE Atlas Platform for Measurement Research. *ACM CCR*, 45(3):35–42, 2015.
4. H. Ballani and P. Francis. Towards a Global IP Anycast Service. In *ACM SIGCOMM*, pages 301–312, 2005.
5. H. Ballani, P. Francis, and S. Ratnasamy. A Measuremnet-based Deployment Proposal for IP Anycast. In *ACM IMC*, pages 231–244, 2006.
6. R. Bellis. Researching F-root Anycast Placement Using RIPE Atlas. https://labs.ripe.net/, 2015.
7. P. Boothe and R. Bush. Anycast Measurements Used to Highlight Routing Instabilities. NANOG 34, 2005.
8. N. Brownlee, kc claffy, and E. Nemeth. DNS Root/gTLD Performance Measurement. In *USENIX LISA*, pages 241–255, 2001.
9. N. Brownlee and I. Ziedins. Response Time Distributions for Global Name Servers. In *PAM*, 2002.

10. R. Bush. DNS Anycast Stability: Some Initial Results. CAIDA/WIDE Workshop, 2005.
11. CAIDA. Skitter. http://www.caida.org/tools/measurement/skitter/.
12. M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *ACM IMC*, pages 313–326, 2013.
13. M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the Performance of an Anycast CDN. In *ACM IMC*, pages 531–537, 2015.
14. S. Castro, D. Wessels, M. Fomenkov, and K. Claffy. A Day at the Root of the Internet. *ACM CCR*, 38(5):41–46, 2008.
15. D. Cicalese, J. Augé, D. Joumblatt, T. Friedman, and D. Rossi. Characterizing IPv4 Anycast Adoption and Deployment. In *ACM CoNEXT*, 2015.
16. D. Cicalese, D. Joumblatt, D. Rossi, M.-O. Buob, J. Augé, and T. Friedman. A Fistful of Pings: Accurate and Lightweight Anycast Enummeration and Geolocation. In *IEEE INFOCOM*, pages 2776–2784, 2015.
17. L. Colitti. Effect of anycast on K-root. 1st DNS-OARC Workshop, 2005.
18. DNS Root Servers. `http://www.root-servers.org/`.
19. X. Fan, J. Heidemann, and R. Govindan. Evaluating Anycast in the Domain Name System. In *IEEE INFOCOM*, pages 1681–1689, 2013.
20. X. Fan, E. Katz-Bassett, and J. Heidemann. Assessing Affinity Between Users and CDN Sites. In *TMA*, pages 95–110, 2015.
21. M. Fomenkov, kc claffy, B. Huffaker, and D. Moore. Macroscopic Internet Topology and Performance Measurements From the DNS Root Name Servers. In *USENIX LISA*, pages 231–240, 2001.
22. Google Public DNS. `https://developers.google.com/speed/public-dns/`.
23. J. H. Kuipers. Analysing the K-root Anycast Infrastructure. `https://labs.ripe.net/`, 2015.
24. 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 *IFIP/IEEE NOMS*, pages 862–865, 2010.
25. T. Lee, B. Huffaker, M. Fomenkov, and kc claffy. On the problem of optimzation of DNS root servers' placement. In *PAM*, 2003.
26. J. Liang, J. Jiang, H. Duan, K. Li, and J. Wu. Measuring Query Latency of Top Level DNS Servers. In *PAM*, pages 145–154, 2013.
27. Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and kc claffy. Two Days in the Life of the DNS Anycast Root Servers. In *PAM*, pages 125–134, 2007.
28. B. Palsson, P. Kumar, S. Jafferalli, and Z. A. Kahn. TCP over IP Anycast – Pipe dream or Reality? `https://engineering.linkedin.com/`, 2015.
29. J. Pang, J. Hendricks, A. Akella, R. D. Prisco, B. Maggs, and S. Seshan. Availability, Usage, and Deployment Characteristics of the Domain Name Server. In *ACM IMC*, pages 1–14, 2004.
30. C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service. RFC 1546, 1993.
31. RIPE NCC. Dnsmon. web site `https://atlas.ripe.net/dnsmon/`, 2015.
32. RIPE NCC Staff. RIPE Atlas: A global Internet measurement network. *The Internet Protocol Journal*, 18(3):2–26, Sept. 2015.
33. Rootops. Events of 2015-11-30. Technical report, Root Server Operators, 2015.
34. S. Sarat, V. Pappas, and A. Terzis. On the use of Anycast in DNS. In *ICCCN*, pages 71–78, 2006.
35. R. d. O. Schmidt, J. Heidemann, and J. H. Kuipers. Anycast latency: How many sites are enough? Technical Report ISI-TR-2016-708, USC-ISI, May 2016.
36. N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, pages 113–124, 2003.
37. F. Streibelt, J. Böttger, N. Chatzis, G. Smaragdakis, and A. Feldman. Exploring EDNS-Client-Subnet Adopters in your Free Time. In *ACM IMC*, pages 305–312, 2013.
38. A. Toonk. How OpenDNS achieves high availability with anycast routing. `https://labs.opendns.com/`, 2013.
39. S. Woolf and D. Conrad. Requirements for a Mechanism Identifying a Name Server Instance. RFC 4892, 2007.