Anycast Latency: How Many Sites Are Enough?

ISI-TR-2016-708, May 2016

Ricardo de O. Schmidt
University of Twente
r.schmidt@utwente.nl

John Heidemann
USC/ISI
johnh@isi.edu

Jan Harm Kuipers
University of Twente
j.h.kuipers@student.utwente.nl

ABSTRACT

Anycast is widely used today to provide important services including naming and content, with DNS and Content Delivery Networks (CDNs). An anycast service uses multiple *sites* to provide high availability, capacity and redundancy, with BGP routing associating users to nearby anycast sites. Routing defines the *catchment* of the users 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 must 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 worldwide vantage points (VPs) in RIPE Atlas. (Given the VPs uneven geographic distribution, we evaluate and control for potential bias.) Key results of our study are to 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 nodes. 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 high-level goals can be limited by bottlenecks in underlying resources: server load, network throughput, latency, and network reliability between the user and server. 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 [13] and IP anycast [1, 33] (or their combination [14, 31]). IP anycast is necessary for DNS service replication, where it is used by many large domain operators, including most root servers, top-level domains, and many large companies, and public resolvers [24, 41]. IP anycast is also used by several web CDNs (Bing, CloudFlare, Edgecast). DNS-based redirection is also widely used (for example, by Akamai, Google, and Microsoft, sometimes in conjunction with IP anycast), but this paper 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*. In the optimal case, users are associated with the nearest site. 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 [29], studies of routing have shown that actual network topology can vary [39], and recent observations by DNS operators have shown that the mapping can be unexpectedly chaotic [7, 25].

Although anycast is widely deployed and critical to multiple network services, prior studies of its effectiveness to reduce latency have been limited—although they identified surprising complexity, they did not explore root causes, optimal possible performance, and their relationship.

The **first contribution** of this paper is to carry out a measurement study of real-world anycast deployments to observe the latency they provide to clients. We study four distinct anycast services, C-, F-, K- and L-Root, each providing part of the Root zone of the Domain Name System. These services are operated by four or-

[1] The term anycast *instance* usually refers to what we call a site, but it can also refer to specific servers at a site. Because of this ambiguity we avoid that term in this paper.

ganizations and encompass different sizes and design decisions, together consisting of more than 240 anycast sites. We observe these services from more than 7,900 vantage points around the world using RIPE Atlas [34, 36]. Our examination focuses on the effects of anycast on latency of DNS queries; although we believe our results generalize to the use of IP anycast for other applications such as anycast CDNs.

This paper focuses on anycast *latency*. We examine latency because latency reduction is a core motivation for millions of dollars in capital and operational expenses, with examples including Google's 2013 buildout to thousands of locations [13], gradual expansion of Root DNS anycast to more than 500 sites [19], and it is central to web and media content distribution for dozens of companies. Nevertheless, we recognize that anycast serves many purposes: in addition to reducing latency between service and users, it can be used to distribute load, improve resilience to denial-of-service attacks, and also to support policy choices. 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).

The **second contribution** of this paper is to provide the first systematic study of the effects of anycast on service 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 (§ 3.4), and does changing them help (§ 3.5)? With these questions resolved, we return to our key contribution and show that a modest number of well-placed anycast sites can provide nearly as good performance as many, but more sites improve the tail of the performance distribution (§ 3.6).

Our **final contribution** is to develop a new measurement methodology necessary to answer these questions. We show how measurements of anycast service addresses can be combined with measurements of their unicast addresses to estimate *optimal* possible performance. Prior work measured only observed anycast performance, and thus could not evaluate optimality and were limited in their ability to estimate routing distortion. Our approach uses site location and unicast addresses publicly provided by the service operators, so our approach does not require approximations.

Our dataset is publicly available at `http://traces.simpleweb.org/`.

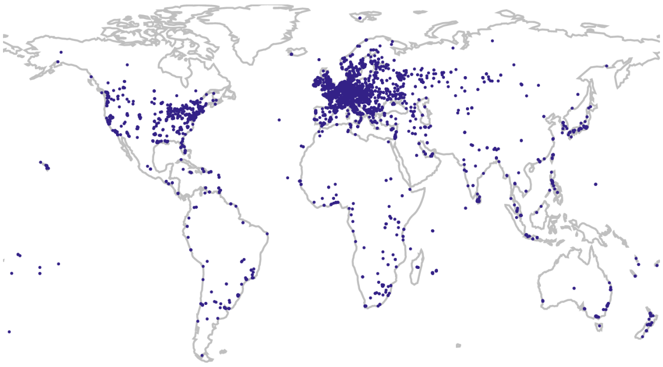


Figure 1: Locations of more than 7,900 vantage points we use from RIPE Atlas.

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.

Specifically, we use the RIPE Atlas infrastructure as VPs to study the C, F, K and L Root DNS services, measuring with pings and DNS CHAOS queries.

We are not the first to examine anycast using pings and CHAOS queries: prior studies examined F-root [7] and K-root [25] from RIPE Atlas, and other studies enumerated all roots and PCH (Packet Clearing House) from PlanetLab [20]. To our knowledge, however, we are the first to measure latency to *all anycast sites* from all VPs, an important step to enable evaluation of optimality and policy questions in § 3.

Measurement sources: Our vantage points (*VPs*) are more than 7,900 probes (measurement devices) in the RIPE Atlas measurement framework [34, 36]. Figure 1 shows the locations of all vantage points that we use: these cover 174 countries and 2927 ASes. RIPE is based in Europe, and there are far more RIPE probes inside Europe than elsewhere. Therefore, the geographic distribution of our vantage points does not match that of the overall Internet population. We maximize coverage by using all RIPE Atlas probes that are available at each measurement time; the exact set varies slightly over our experiments.

We will show later (§ 3.3) that this skew strongly affects the *specific, quantitative* latencies we observe, favoring sites with more anycast sites in Europe. However, it **does not** affect our *qualitative* results about the role of number of anycast sites and the effects of routing policies.

The exact number of VPs that see each service vary, as shown in Table 1. Our measurements each take place over several days and were carried out at different times

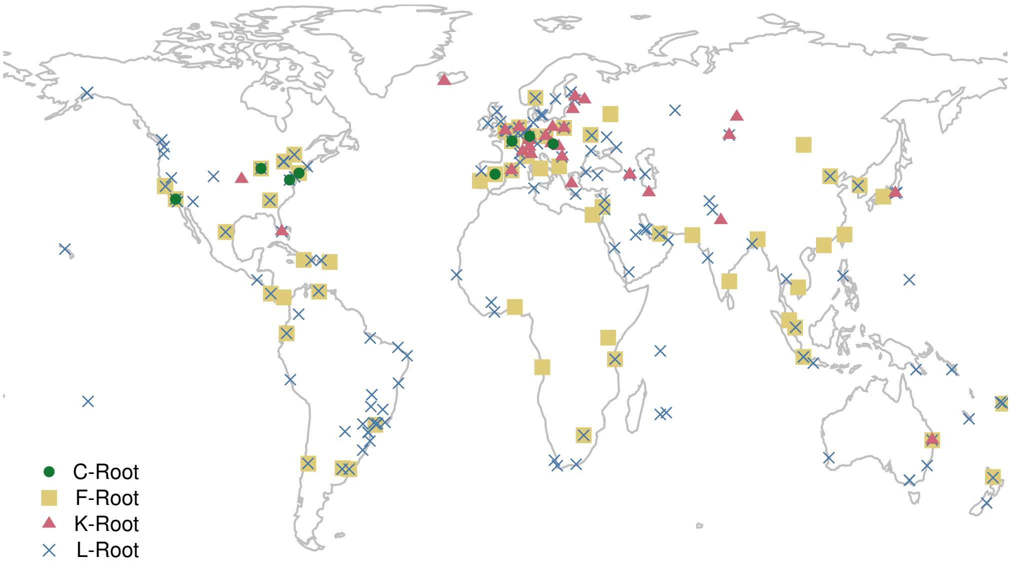


Figure 2: Locations of sites for each service: C, green circles; F, gold squares; K, red triangles; and L, blue crosses.

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

Table 1: Summary of each root service (left), how many VPs are used to study each, and the catchments we observe, reporting hits and mishits and their latencies. (Number of VPs vary due to measurement at different times; and number of sites is as of measurement date.) K-Root was measured twice.

over 2015 and 2016; this variation depends on VP availability. We do not believe it biases our results.

Measurement targets: We select as targets of our study four operational anycast services: the C-, F-, K- and L-Root DNS services [19]. These four real-world, operational anycast services are each optimized by its operator to meet the goals of its organization. They are diverse, with a range of sizes (with C small, F and K mid-sized, and L numerous). Their routing policies also vary: all C and L sites are global (available to all), while many K and most F sites are local, with service limited to specific Autonomous Systems. For our evaluation of optimal latency (§ 3), an essential point is that all of these services make public both their anycast service address and the unicast addresses for each anycast site. We get this information from CHAOS queries, and confirm against `www.root-servers.org`.

Locations of anycast sites for each service are given in Figure 2. C operates only in North America and Europe; all others have sites around the world.

We measured K Root twice: in 2015 (K) and in 2016 (NK—*New K*). This is because after our first measurement K changed its anycast policies. These changes and their implications are discussed in § 3.5.

Measuring anycast catchments: We determine the *anycast catchment* seen by each vantage point using DNS CHAOS queries [43] to the anycast service address of each target. The reply to a CHAOS query contains a string that uniquely identifies the anycast site for that vantage point 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 anycast site. An example of CHAOS reply for C Root is `lax1b.c.root-servers.org`, where `lax` gives the geographic location of the replying anycast site, and `1b` identifies the replying server within the site. Since we focus on anycast deployment, in this work we do not consider potential differences between servers of an anycast site. (Note that by *CHAOS queries* we mean queries to certain TXT records such as `hostname.bind`, in the CLASS

CHAOS that return a string identifying the answering sever [43].)

Measuring latency: For each target service, we measure *latency* to both the public anycast service address and the unicast address of each site using ICMP ECHO requests (pings). To suppress noise in individual pings, we send multiple pings to each site and report the 10th-percentile value as the actual latency. The exact number of pings varies; from each vantage point to each root site, we send on average 36 pings for C Root, 89 pings for K Root, and 30 pings for F and L Root. These variations are caused by dynamics on the RIPE Atlas framework, based on limitations on availability of VPs and measurement scheduling.

3. OBSERVATION AND FINDINGS

Our goal is to understand how many anycast sites are "enough" for good performance. We first build up on the basic question: defining possible performance, exploring how users associate with anycast sites, the effects of location and measurement bias, and local policies. These allow us to understand how many anycast sites are needed and who sees poor latency given that number.

3.1 Does anycast give good absolute performance?

We first look at absolute latency seen from our vantage points for each anycast service.

Figure 3 shows the distribution of latency seen from each VP to the C- (green), F- (gold), K- (red) and L-Root (blue) services. It reports the *actual* RTT to each VP's BGP-assigned anycast site. We see that *all letters provide low latency to most users*: for C and K Root, half of the VPs see a RTT of 32 ms or less, L's median RTT is 30 ms, and F is 25 ms.

Second, we see 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 usually no more than 30 ms different between any letters at any point of the distribution, and within 15 ms if we ignore F-Root. (At the tail of the distribution however this difference increases up to 135 ms—not visible in Figure 3.) This result is quite surprising since there is a huge difference on the sizes of the anycast deployments of these four anycast deployments, from 8 to 144 sites.

Finally, while more anycast sites usually provide slightly better performance, this trend does not always hold. We see that F's median latency is lower than L's (25 ms vs. 30 ms), even though it has fewer anycast sites (58 vs. 144 sites). This difference is perhaps due to careful engineering of F's anycast routes explicitly using RIPE Atlas for debugging [7]. We do not know if this difference holds when observed from other sites, but it strongly suggests that *careful engineering and route management is important.*

3.2 Do users get the closest anycast site?

Anycast relies on BGP to match users to sites, but BGP only approximates shortest-path routing. We next explore how often our vantage points see the closest anycast site compared to *mishits* when they see some other site, and how much a mishit affects latency.

Figure 4 shows the *optimal possible* performance (dotted lines), based on unicast routing, ignoring anycast routing policies and catchments. (For comparison purposes, Figure 4 also shows the *actual* performance seen across all VPs—solid lines.) Comparing the values C-root (the green lines), shows that C-root's actual service is very close to optimal (the green solid and dashed lines nearly overlap). C does well because it has only a few, geographically distributed sites, and its nodes are all global, without routing policy limitations—both simplifying cases that make routing optimization for latency "easy".

By contrast, larger anycast deployments show latency inefficiencies both because more sub-optimal choices are available, and because these services have some or many local nodes that place policy limitations on routing. Focusing on the median RTT for F-, K- and L-Root (we consider the distribution's tail later § 3.6.2), we see that routing freedom would improve latency by 16 ms, 19 ms and 14 ms, respectively. For F-, K- and L-Root this represents an improvement of 36%, 40% and 53% of their actual performance (25, 32 and 30 ms respectively). (Of course, routing limitations may be a condition of site deployment. We wish to understand the potential optimal, even if it may be unrealizable.)

To better understand the potential cost of policy routing, we next focus on *mishits*: VPs that are sent to sites that are not closest. Table 1 shows how often mishits occur. The number of mishits naturally follow the number of sites, since more sites give more opportunity to be routed to a more distant one (*i.e.,* not optimal site). In addition, services with more sites often have local-only sites, where routing policy limits access.

Returning to performance, Figure 5 shows the latencies seen by mishits (dashed lines) to the four anycast services (RTT distribution for all actual hits is plotted as solid lines for comparison). We see that the impact of mishit is worse for C Root. Missing your nearest site often has a serious cost, with median latency of 40 ms or higher for all letters we show (from Table 1: F, 39 ms; K, 43 ms; L, 47 ms; and C, highest at 61 ms).

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 routing policies). Figure 6 shows the distribution

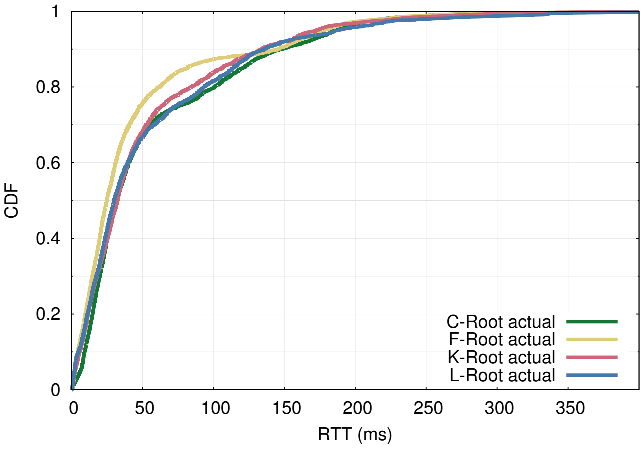


Figure 3: Distribution of RTT to all actual hit for C- (green), F- (gold), K- (red) and L-Root (blue).

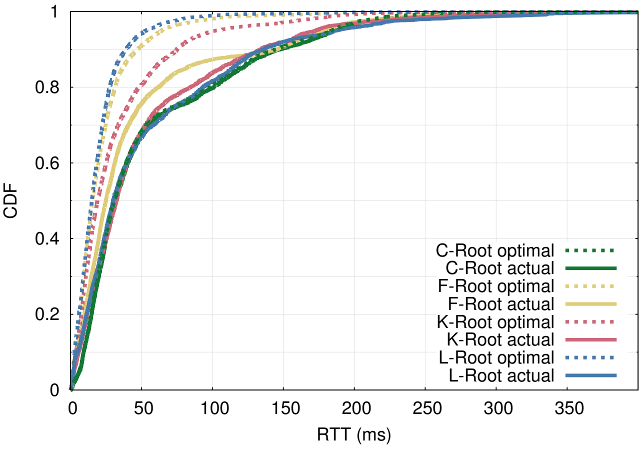


Figure 4: Distribution of RTT to C- (gree), F- (gold), K- (red) and L-Root (blue): *optimal* hit ignoring BGP (dotted) compared to all *actual* hit (solid).

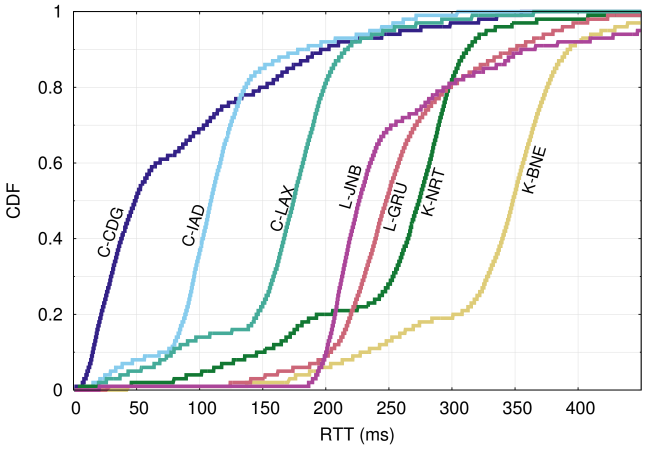


Figure 7: Distribution of RTTs to single locations around the world, as suggested by selected C and K root sites.

of the mishit penalty to all four anycast services. The median penalty for F, K, and L Roots are 15, 18, and 24 ms, respectively, with C root much smaller at only 5 ms. With many local sites, F and K are often constrained by routing policy, something we study later in § 3.4. With L Root's many sites, there are many opportunities to mishit (76% of VPs mishit—Table 1) and, although the alternative selection is often nearby, the mishit penalty for L Root is still the highest.

Surprisingly, C-Root's few sites also have the lowest penalty of mishitting. We believe this low penalty is because C's sites are well connected and relatively close to each other (in the U.S. or Europe), so missing the closest often results in finding another on the same continent, incurring little additional delay. (The last columns of Table 1 show the 25, 50 and 75th percentiles of the distribution of mishit penalties to all four anycast services.)

3.3 Effects of Anycast Location on Latency & Observation Bias

Current anycast infrastructure provides good actual latency (§ 3.1) and near optimal results for a given infrastructure (§ 3.2). We next consider how the location of anycast sites affects observed latency, showing the effects of speed-of-light communication delay, and the locations of our vantage points (showing that our vantage points are strongly weighted to Europe).

3.3.1 Single locations

We first consider: what if a single site provided service for all of the world? Figure 7 shows the latency from *all* VPs to seven different sites operated by several services. We see huge variation on median RTTs for each site (from 50 to 350 ms), and each site shows a fraction of VPs with extreme latency.

Before considering what this data says about anycast, we must observe that it shows *a strong bias of our vantage points in favor of Europe.* This bias occurs because most RIPE probes are in Europe (Figure 1), and shows in the European site (C-CDG) having lowest median latency, and with the distribution of latencies for each site dominated by the speed-of-light delay to Europe (for example, L-GRU in São Paulo is about 200 ms from Europe, and K-BNE in Brisbane, Australia is the most distant). We highlight this bias here; it must be considered in all of our observations. This bias reflected in measurement tools based on RIPE Atlas, such as DNS-

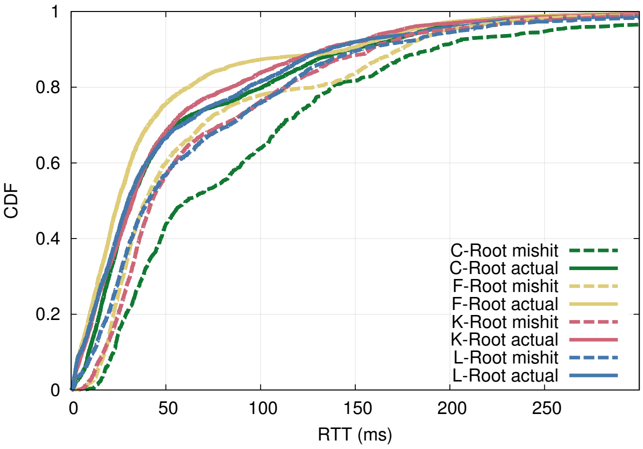


Figure 5: Distribution of RTT for *mishits* (dashed) to C- (green), F- (gold), K- (red) and L-Root (blue); all actual hit (solid) for comparison.

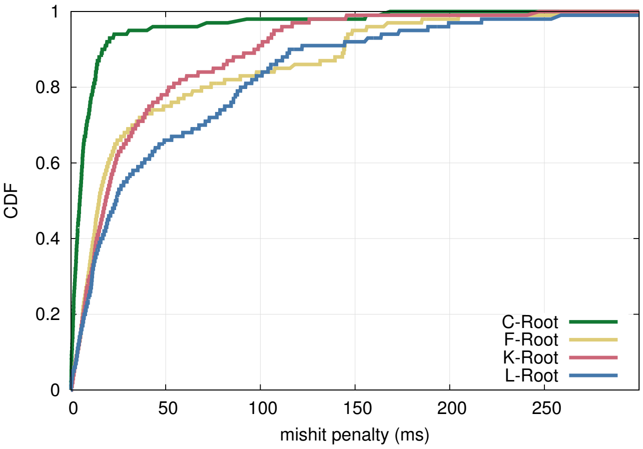


Figure 6: Distribution of penalty latency—mishit RTT minus optimal RTT—for *mishits* of C- (green), F- (gold), K- (red) and L-Root (blue).

MON [35], and that RIPE Atlas does not represent all global traffic [37].

While the bias colors the specific median latencies, the data strongly shows that *no single location can provide equally low latency to the global Internet*—all sites show many VPs with latencies exceeding 100 ms. This result is well known and is one motivation for distributing the pre-anycast roots to organizations around the world. It also shows the importance of the common practice where recursive resolvers favor servers with low latency (when given a choice). Selection of a letter with a single site (such as B Root) or a few "regional" sites (such as H Root) will not provide best possible performance from all locations.

3.3.2 Choice of multiple locations

We next show that site location is important for anycast services by simulating an anycast service with one to four sites. We select locations drawn from C Root and determine latency for our simulated service by assuming all clients choose the closest site using real-world observations. While we know that clients sometimes choose non-closest sites, we have shown that this effect is very small (§ 3.2).

Figure 8 compares four subsets of C-Root's infrastructure to C-Root optimal. The subsets begin on the right using a single-site in Los Angeles (LAX), then add C-Root sites going mostly eastward, with Chicago (ORD), Washington, DC (IAD), and 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-node configuration). This trend reflects speed-of-light from our European VPs to the U.S., with better latency coming as sites closer to Europe are added. We also see that the addition of New York (JFK) has almost no improvement over three sites with Washington (IAD); Washington is almost as close to Europe as New York.

3.3.3 Geographic distribution of site location

The west-to-east selection of C-Root sites in Figure 8 is the worst possible node selection when most VPs are in Europe. We can minimize speed-of-light delays by maximizing the geographic distance between anycast sites, starting with the site closest to the majority of our observers.

Figure 9 shows the analysis of theoretical anycast deployments drawn from C's sites chosen to maximize inter-site distance, starting in Europe. We start with a site in Paris (CDG), close to the majority of our VPs in Europe, and with a tail of VPs elsewhere in the world—this configuration is within 20% of optimal (as defined by all of C-Root's 8 sites). We then add U.S. west and east coasts (LAX and JFK respectively), then Frankfurt (FRA), each pulling the distribution closer to optimal, particularly in the tail. This data supports 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 the speed-of-light (in fiber) limit. Although both network topology and routing policies can cause "close" in the network to diverge from geographic proximity [39], geographic dispersion can promote dispersion in network topology. We next consider policy effects on latency.

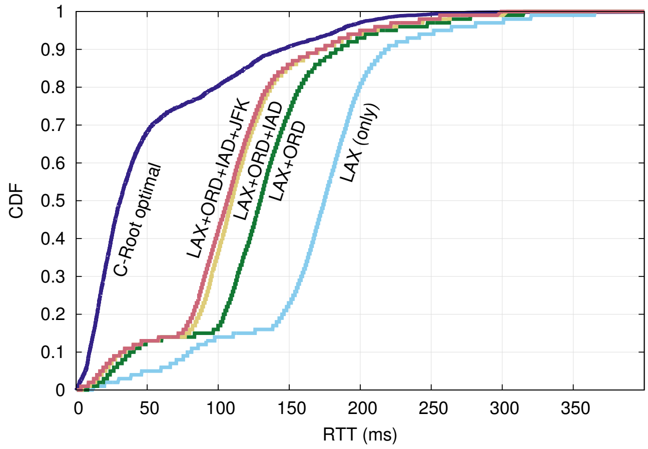


Figure 8: Distribution of RTTs to an anycast service with 1 to 4 anycast sites, simulated from U.S.-based C-Root sites from west to east.

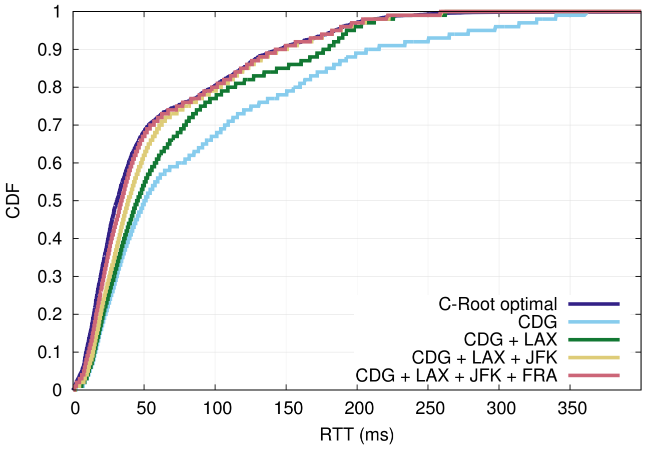


Figure 9: Distribution of RTTs to an anycast service with 1 to 4 anycast sites, simulated using C-Root sites chosen to maximize geographic separation.

3.4 Do local anycast policies hurt performance?

Different root letters have very different routing policies. Of those we study, C and L place no restrictions on routing, while about half of K-Root and most F-Root sites are local, limiting routing to the direct or adjacent AS (Table 1).

We have shown that all letters have similar distributions of latency (Figure 3), suggesting that routing policy does not greatly distort latency. However, we next look at F- and K-Root, the services with many local sites, to confirm this overall trend.

To focus on routing policy, Figure 10 examines the subsets of F- and K-Root where VPs *hit* (route to) either global (green) or local (blue) sites. Local services is achieved by announcing the anycast prefix with BGP options `NO_EXPORT` or `NOPEER`, limiting propagation of that route to the immediate AS. We do not measure routing directly, but use operator provided lists of each site's policy.

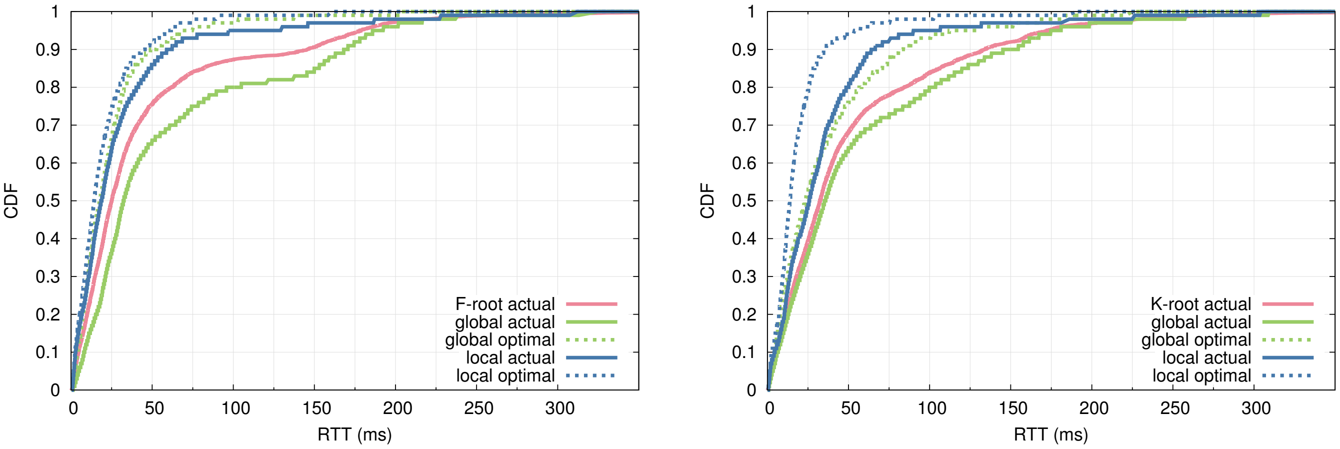
We first observe that most sites actually use global sites: 67% of VPs for F Root, and 75% for K Root (Table 2).

Figure 10 breaks out performance based on VPs that access local and global sites. VPs hitting local sites (solid blue line) have much lower latency than those hitting global sites (solid green line). Figure 10a shows that for F Root 85% of VPs hitting a local site see a latency ≤50 ms, and only 5% see a latency ≥100 ms. For those VPs hitting a F-Root global site only 65% see a RTT ≤50 ms, while 20% see a latency ≥100 ms. K-Root shows a similar trend (Figure 10b), although VPs hitting a K-Root global site see a slightly lower RTT at the distribution tail, likely because K Root has more global sites (and so more close sites) than F Root.

We would like to understand *how* and *how much* these policies affect the performance of F and K Root's infrastructure: how often would VPs that hit global sites prefer local sites? Table 2 breaks each vantage point that mishit F and K Root (those that did not hit the closest anycast site) into four categories: when the actual chosen anycast site and the optimal site are either global (G) or local (L).

In this table policy mismatches occur in the italicized G-L and L-G categories (rows 2 and 4 respectively), where either a VP cannot choose a closer local site (G-L) or its route prefers a local site over a nearer global site (L-G).

For **G-L mismatches** the VPs are prevented from accessing a local node by its policy. For both F and K Root, G-L mismatches are the most common among mishits: 58% of VPs hitting a F Root's global site would prefer to reach a local site, and for K Root 42% of VPs hitting global sites would prefer a local one. By “solving” G-L mishits a significant improvement on latency could be achieved for F and K Root (compare solid and dotted green lines in Figure 10). The policy that causes G-L mismatches may have been a condition of deploying the anycast site there (some ISPs are happy to host an anycast site to improve service to their customers, but do not want additional external traffic), or due to limited network or server capacity, encouraging hosts of local-only sites to broaden their routing policies would improve service in this case. Ultimately this policy is the choice of the anycast service and its hosting ISP, but we suggest an alternative to improve performance (and reduce the number of global mishits) could be to deploy a twin global site with each local site. This proposal would guarantee local service while also improving global performance.



(a) F Root

(b) K Root

Figure 10: Distribution of latency to K and F Root, as actually seen from all VPs (red), and for VPs that hit global (green) and local (blue) sites, with both their actual and optimal distributions.

	actual hit	closest	F Root count			K Root count		
G-G	global	global	308	(9%)	[67%]	1246	(33%)	[75%]
G-L	*global*	*local*	2058	(58%)		1621	(42%)	
L-L	local	local	1025	(29%)	[33%]	655	(17%)	[25%]
L-G	*local*	*global*	148	(4%)		301	(8%)	
	total vantage points		3539	100%		3823	100%	

Table 2: The influence of local routing policies: mishits for F and K Root.

For **L-G mismatches** a local anycast site captures traffic that would be better sent elsewhere. This case may be an ISP with the goal of reducing external traffic, or it may be the imperfections in shortest-path typical to BGP routing. For both F and K Root, L-G mismatches do not represent a large portion of mishits with 4% and 8% of VPs respectively.

L-L mismatches are also likely to be a consequence of routing policies: VPs hitting a local anycast site are prevented from reaching their preferred site because it is local to a different network. L-L mismatches consists of a large amount of F Root mishits (29%). That is because F Root is composed mostly by local sites and, hence, the optimal hit for a given VP is likely to be one of F's local sites.

Finally, **G-G mismatches** in Table 2 happen when VPs are routed to a global anycast site while a different global site is the optimal hit. This mismatch is not caused by routing policies, but are likely consequences of imperfections on BGP routing, which are difficult to fix due to the complexity of peering agreements. The G-G mismatches are more common for K Root with 33% of mishits because of K's higher number of global sites.

The cases discussed in this section suggest the reasons for the divergence from optimal performance for F and K Root: *many local anycast sites are intentionally not accessible*, preventing the anycast service to achieve a much better performance latency wise.

3.5 Does relaxing routing policy help?

Our evaluation of K-Root in 2015-11 occurred when about half of its sites were local-only, and we hypothesized these policy choices contributed to the gap between K's actual and optimal latency (Figure 4).

However, in a gradual process that started in early 2015 and was concluded in early 2016, K-Root made all but one of its sites global. (In this process they also added three new sites.) From informal talks with RIPE staff (K's operator) we learned that this change was motivated by the fact that: (i) enforcing NO_EXPORT is almost impracticable because peers mostly ignore such request, and (ii) in some cases this request is too rigid and unnecessarily limits the propagation of the anycast prefix. This change provides a natural experiment to evaluate our hypothesis: relaxing routing constraints should allow users to reach closer sites, reducing overall latency. We evaluated K-Root's new anycast system in 2016-04 to see what really happens.

Figure 11 compares K-Root in 2015 with local routing and in 2016 without (K vs. NK). We see that routing policy did *not* greatly affect K-Root's latency. The median RTT is almost the same: falling only 2 ms from 32

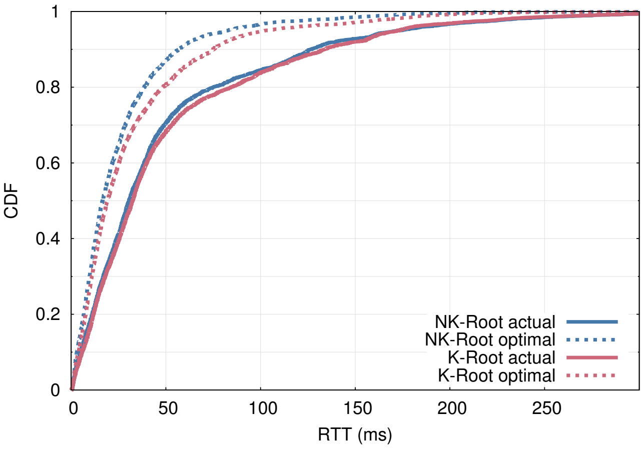


Figure 11: Distribution of RTT to NK- (blue) and K-Root (red): *optimal* ignoring BGP (dotted) compared to all *actual* (solid).

to 30 ms. The largest reduction in latency is only 10 ms. The largest change is that the tail of the optimal CDF (from 0.7 to 1) is reduced, but little change is seen in actual latency.

Our evaluation of mishit rates in Table 1 shows the reason there is little improvement: mishits show almost no change, falling from 43% to 41%. We conclude that altering policy alone does not address the gap between actual and optimal. Instead, it seems that a case-by-case analysis of anycast deployment and routing policies is necessary to reduce latency in large anycast deployments, as was carried out by F-Root in 2015 [7]. Possible future work is to provide tools to automate such optimization of deployments.

3.6 Do many sites help?

We have seen that the number, location and routing policies largely affect latency that our VPs observe. Root letter operates services of very different sizes: few sites (A and C at 5 and 8), a medium number (F, I, and K with 36 to 58), or many (D, J, and L, each with about 100 or more). We now ask: does having many sites improve latency? And we look at this question in two ways: first we examine existing deployments, then we look at the tail of the distribution and what users have the poorest latency.

3.6.1 Comparing numbers of sites

We compared C, F, K and L Root in Figure 3. A key result is that C, K and L provide roughly similar latency across our vantage points—compared to C the additional sites in K and L do not even show improvement to the median latency (32 ms for C and K, and 30 ms for L Root). We also saw that the specific locations of sites can have different levels of improvement on latency: the strategically positioned sites of C Root result in a performance as good as K and L, which have respectively 4 and 18 times more sites than C.

While median latency is quite similar across C, K and L, the largest difference is in the tail of the distribution, from 70th to 90th percentiles of the distribution. We next examine the tail of this distribution.

3.6.2 Who sees poor latency?

Our results in § 3.3 show that speed-of-light delays are a primary influence on latency. To evaluate the tail of the latency distribution, we next report mean latency by country for all countries with at least 5 Atlas probes. Countries with fewer probes tend to show worse latency than those we show, but we wish to exclude outliers. We do not know the local network of each VP, and it would be unfeasible to treat each probe individually, so we do not account for poor “last-mile” connections [4].

Figure 12 shows the latency for all the countries with at least 5 probes. This data confirms our hypothesis: the countries with poorest latency are mostly in Africa, Oceania and Asia, with a few from South America. For C (Figure 12a) and K Root (Figure 12c) this result was expected, since they do not have sites widely spread worldwide. However, the same pattern is observed for F and L Root, suggesting that poor connectivity is a major problem in those regions. The larger and more distributed deployments of F and L Root seem to help on improving the latency of certain vantage points in the worst regions, as shown by the quartiles in the left-half of Figure 12b and Figure 12d. That is, this seems to suggest that although F and L Root have sites all over the world, the performance for clients in areas with “poorer” connectivity strongly depends on which network they are.

4. RELATED WORK

The DNS root server have been extensively studied in the past. CAIDA’s measurement infrastructure `skitter` [12] has enabled several early studies on DNS performance [9, 22, 10, 27]. In 2004, Pang *et al.* [32] used a mix of active probing and log analysis to show that although DNS servers were highly available, only few of them were being used by a large fraction of users. Other following work also studied the performance of DNS, mainly focused on latency measurements between clients and servers [18, 6, 38]. The DNS CHAOS have also been used to study client-server affinity [38, 8]. Using a different approach, Liu *et al.* [30] used geolocation of clients to estimate RTT, and others evaluated the effect of route changes on the anycast service [5, 11]. Already in 2013, Liang *et al.* [28] used open resolvers to measure the RTT from the DNS root and major gTLDs, and showed that latency can be up to 6 times worse in poorly served regions. Finally, Bellis [7] carried out a comprehensive assessment of latency in F Root’s anycast. With 58 sites and a mix of local and global routing he found that fixing faulty announcements improved performance. Other interesting work [15, 26] used large and long-term datasets to characterize the deployment, usage and assess the performance of the root servers. Results in this work show that the expansion of the anycast infrastructure at the root level helped to improve the performance of the whole system by reducing the RTT between server and clients.

Calder et al. examined the choice of anycast or LDNS for redirection to CDN services [14]. We both measure user latency, but they start with a given anycast infrastructure (Microsoft Bing) and the mechanism for user mapping, while we instead vary the size of the infrastructure and use only anycast mapping.

Our work differs from these prior studies in methodology and analysis. Our methodology builds on prior studies of latency and catchment, but unlike prior work we add concurrent probes to *all* sites (via their unicast addresses) to allow comparison to optimality. In addition, we need not to estimate geolocation since our VPs and targets both provide accurate geographic information. Our analysis differs from prior work in two ways. First, we go beyond evaluating the observed anycast catchment, and instead use measurements to all sites to define the theoretically optimal performance for a given anycast infrastructure. Second, we use complete latency information to evaluate alternative, theoretical anycast infrastructures where nodes are placed in different geographic regions. With this additional analysis we evaluate how design choices affect potential future anycast deployments, in addition to how it is used today.

Complementing our work are studies that enumerate and characterize content delivery services that use IP anycast or other techniques. Two researchers focus on YouTube's CDN: Torres *et al.* [42] used datasets collected at few observation points with constraint-based geolocation, and Adhikari *et al.* [2] used open DNS resolvers and geolocation databases to understand dynamics and operational strategies of YouTube. Calder *et al.* [13] used EDNS-client-subnet (ECS) and latency measurements to identify and characterize Google's serving infrastructure. They show Google's growth in 2013 into large and small ISPs and suggest it was to reduce user-to-service latency. Streibelt *et al.* [40] also used ECS measurements to, among others, study user to server mappings in the anycast infrastructure of major ECS adopters such as Google, Edgecast and CacheFly. Fan *et al.* [20] used a combination of DNS queries and traceroute measurements to identify and characterize anycast nodes and, among their findings, they showed that up to 72% of all TLDs use anycast. Cicalese *et al.* [17] proposed enumeration and city-level geolocation of anycast services using latency measurements, then used it to characterize IPv4 anycast adoption [16]. In later work [21], Fan *et al.*combined ECS with open DNS resolvers to measure front-ends in Google and Akamai's CDNs. They showed that prefixes are assigned to different clusters, and these reassignments can result in latency shifts of more than 100 ms. Finally, Akhtar *et al.* [3] proposed a statistical approach for comparing the performance of CDNs from active measurements, and Giordano *et al.* [23] used passive traces from a single vantage point to study the popularity of CDNs, showing that up to 50% of web users are served by anycast CDNs during peak hours.

5. CONCLUSIONS

In this paper we have studied four real-world anycast deployments (the C-, F-, K- and L-Root DNS nameservers) to systematically explore the relationship between IP anycast deployment and latency. We developed a new measurement methodology that uses vantage points at more the 7,900 RIPE Atlas sites to observe actual anycast latency, and to compute optimal possible latency for each service. We collected new data for each of these systems in 2015 and revisited K-Root in 2016 to evaluate changes in its routing policies.

Our methodology opens up future directions: although we focused here on latency, we plan to also evaluate other reasons for anycast such as resilience to denial-of-service and load balancing. To complement our current measurement and analysis we are developing an anycast testbed with sites worldwide. We expect to use the testbed to study deployment operational questions experimentally. We also plan to use the measurement strategies we pioneered to develop online monitoring and improve the stability of anycast systems such as the DNS Root zone.

Our central question in this paper, though, is to understand how many anycast sites are "enough". Our methodology and data allow us to untangle several factors to answer this question: we showed that even small deployments can give good absolute performance. Much more important than large numbers of sites is distributed geographic locations of sites. We found that C-, K- and L-Root see the same median RTT (30–32 ms), even K and L having 4 to 18 times more sites than C-Root.

A second critical question is the effects of routing policies, such as whether local-only routing limits optimal performance. We examined routing policy mismatches to estimate these costs. Somewhat surprisingly, we found that presence of local-only routing increases latency only modestly, something confirmed by analysis of K-Root both with and without these policies. Instead, F-Root shows that careful management of specific routing configurations is required to minimize latency and have actual latency closer to optimal.

Finally, we examined the causes of high latency for users in the tail of the distribution. Here we expected the geographic diversity of many sites in F- and L-Root to pay off compared to C and K-Root with little or no footprint in Asia and the southern hemisphere. Surprisingly, we see that anycast sites in these areas are too often unavailable to nearby users, with F- and L-Root still having users with high latencies from non-local sites. Our results suggests that reducing latency requires not only servers in these parts of the world, but also greater local connectivity and careful local routing.

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 valuable technical feedback.

This research is supported by measurements obtained from RIPE Atlas, an open measurements platform 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.

6. REFERENCES

[1] Abley, J., and Lindqvist, K. E. Operation of Anycast Services. RFC 4786, 2006.

[2] Adhikari, V. K., Jain, S., Chen, Y., and Zhang, Z.-L. Vivisecting YouTube: An Active Measurement Study. In *Proceedings of the IEEE INFOCOM* (2012), pp. 2521–2525.

[3] Akhtar, Z., Hussain, A., Katz-Bassett, E., and Govindan, R. DBit: Assessing Statistically Significant Differences in CDN Performance. In *Proceedings of the IFIP Traffic Monitoring and Analysis (TMA)* (2016).

[4] Bajpai, V., Eravuchira, S. J., and Schönwälder, J. Lessons Learned From Using the RIPE Atlas Platform for Measurement Research. *ACM SIGCOMM Computer Communication Review (CCR) 45*, 3 (2015), 35–42.

[5] Ballani, H., and Francis, P. Towards a Global IP Anycast Service. In *Proceedings of the ACM SIGCOMM* (2007), pp. 301–312.

[6] Ballani, H., Francis, P., and Ratnasamy, S. A Measuremnet-based Deployment Proposal for IP Anycast. In *Proceedings of the ACM Internet Measurement Conference* (2006), IMC, pp. 231–244.

[7] Bellis, R. Researching F-root Anycast Placement Using RIPE Atlas. https://labs.ripe.net/, 2015.

[8] Boothe, P., and Bush, R. Anycast Measurements Used to Highlight Routing Instabilities. NANOG 34, 2005.

[9] Brownlee, N., kc claffy, and Nemeth, E. DNS Root/gTLD Performance Measurement. In *Proceedings of the USENIX LISA conference* (2001), pp. 241–255.

[10] Brownlee, N., and Ziedins, I. Response Time Distributions for Global Name Servers. In *Proceedings of the International conference on Passive and Active Measurements* (2002), PAM.

[11] Bush, R. DNS Anycast Stability: Some Initial Results. CAIDA/WIDE Workshop, 2005.

[12] CAIDA. Skitter. http://www.caida.org/tools/measurement/skitter/.

[13] Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., and Govindan, R. Mapping the Expansion of Google's Serving Infrastructure. In *Proceedings of the ACM Internet Measurement Conference* (2013), IMC, pp. 313–326.

[14] Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., and Padhye, J. Analyzing the Performance of an Anycast CDN. In *Proceedings of the ACM Internet Measurement Conference* (2015), IMC, pp. 531–537.

[15] Castro, S., Wessels, D., Fomenkov, M., and Claffy, K. A Day at the Root of the Internet. *ACM Computer Communication Review 38*, 5 (2008), 41–46.

[16] Cicalese, D., Augé, J., Joumblatt, D., Friedman, T., and Rossi, D. Characterizing IPv4 Anycast Adoption and Deployment. In *Proceedings of the ACM CoNEXT* (2015).

[17] Cicalese, D., Joumblatt, D., Rossi, D., Buob, M.-O., Augé, J., and Friedman, T. A Fistful of Pings: Accurate and Lightweight Anycast Enummeration and Geolocation. In *Proceedings of the IEEE INFOCOM* (2015), pp. 2776–2784.

[18] Colitti, L. Effect of anycast on K-root. 1st DNS-OARC Workshop, 2005.

[19] DNS Root Servers. http://www.root-servers.org/.

[20] Fan, X., Heidemann, J., and Govindan, R. Evaluating Anycast in the Domain Name System. In *Proceedings of the IEEE INFOCOM* (2013), pp. 1681–1689.

[21] Fan, X., Katz-Bassett, E., and Heidemann, J. Assessing Affinity Between Users and CDN Sites. In *Proceedings of the 7th IEEE International Workshop on Traffic Monitoring and Analysis* (2015), TMA, pp. 95–110.

[22] Fomenkov, M., kc claffy, Huffaker, B., and Moore, D. Macroscopic Internet Topology and Performance Measurements From the DNS Root Name Servers. In *Proceedings of the USENIX LISA conference* (2001), pp. 231–240.

[23] Giordano, D., Cicalese, D., Finamore, A., Mellia, M., Munafò, M., Joumblatt, D. Z., and Rossi, D. A First Characterization of Anycast Traffic from Passive Traces. In *Proceedings of the IFIP Traffic Monitoring and Analysis Workshop (TMA)* (2016).

[24] Google Public DNS. https://developers.google.com/speed/public-dns/.

[25] Kuipers, J. H. Analysing the K-root Anycast Infrastructure. https://labs.ripe.net/, 2015.

[26] Lee, B.-S., Tan, Y. S., Sekiya, Y., Narishige, A., and Date, S. Availability and Effectiveness of Root DNS servers: A long term study. In *Proceedings of the IEEE Network Operations and Management Symposium* (2010), NOMS, pp. 862–865.

[27] Lee, T., Huffaker, B., Fomenkov, M., and kc claffy. On the problem of optimzation of DNS root servers' placement. In *Proceedings of the International conference on Passive and Active Measurements* (2003), PAM.

[28] Liang, J., Jiang, J., Duan, H., Li, K., and Wu, J. Measuring Query Latency of Top Level DNS Servers. In *Proceedings of the 14th International conference on Passive and Active Measurements* (2013), PAM, pp. 145–154.

[29] Liu, Z., Huffaker, B., Fomenkov, M., Brownlee, N., and kc claffy. Two days in the life of the dns anycast root servers. In *Proceedings of the Passive and Active Measurement Workshop* (Louvain-la-neuve, Belgium, Apr. 2007), Springer-Verlag, pp. 125–134.

[30] Liu, Z., Huffaker, B., Fomenkov, M., Brownlee, N., and kc claffy. Two Days in the Life of the DNS Anycast Root Servers. In *Proceedings of the 8th International conference on Passive and Active Measurements* (2007), PAM, pp. 125–134.

[31] Palsson, B., Kumar, P., Jafferalli, S., and Kahn, Z. A. TCP over IP Anycast – Pipe dream or Reality? https://engineering.linkedin.com/, 2015.

[32] Pang, J., Hendricks, J., Akella, A., Prisco, R. D., Maggs, B., and Seshan, S. Availability, Usage, and Deployment Characteristics of the Domain Name Server. In *Proceedings of the ACM Internet Measurement Conference* (2004), IMC, pp. 1–14.

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

[34] RIPE NCC. RIPE Atlas. web site `https://atlas.ripe.net/`, 2010.

[35] RIPE NCC. Dnsmon. web site `https://atlas.ripe.net/dnsmon/`, 2015.

[36] RIPE NCC Staff. RIPE Atlas: A global Internet measurement network. *The Internet Protocol Journal 18*, 3 (Sept. 2015), 2–26.

[37] Rootops. Events of 2015-11-30. Tech. rep., Root Server Operators, Dec. 4 2015.

[38] Sarat, S., Pappas, V., and Terzis, A. On the use of Anycast in DNS. In *Proceedings of the 15th International Conference on Computer Communications and Networks* (2006), ICCCN, pp. 71–78.

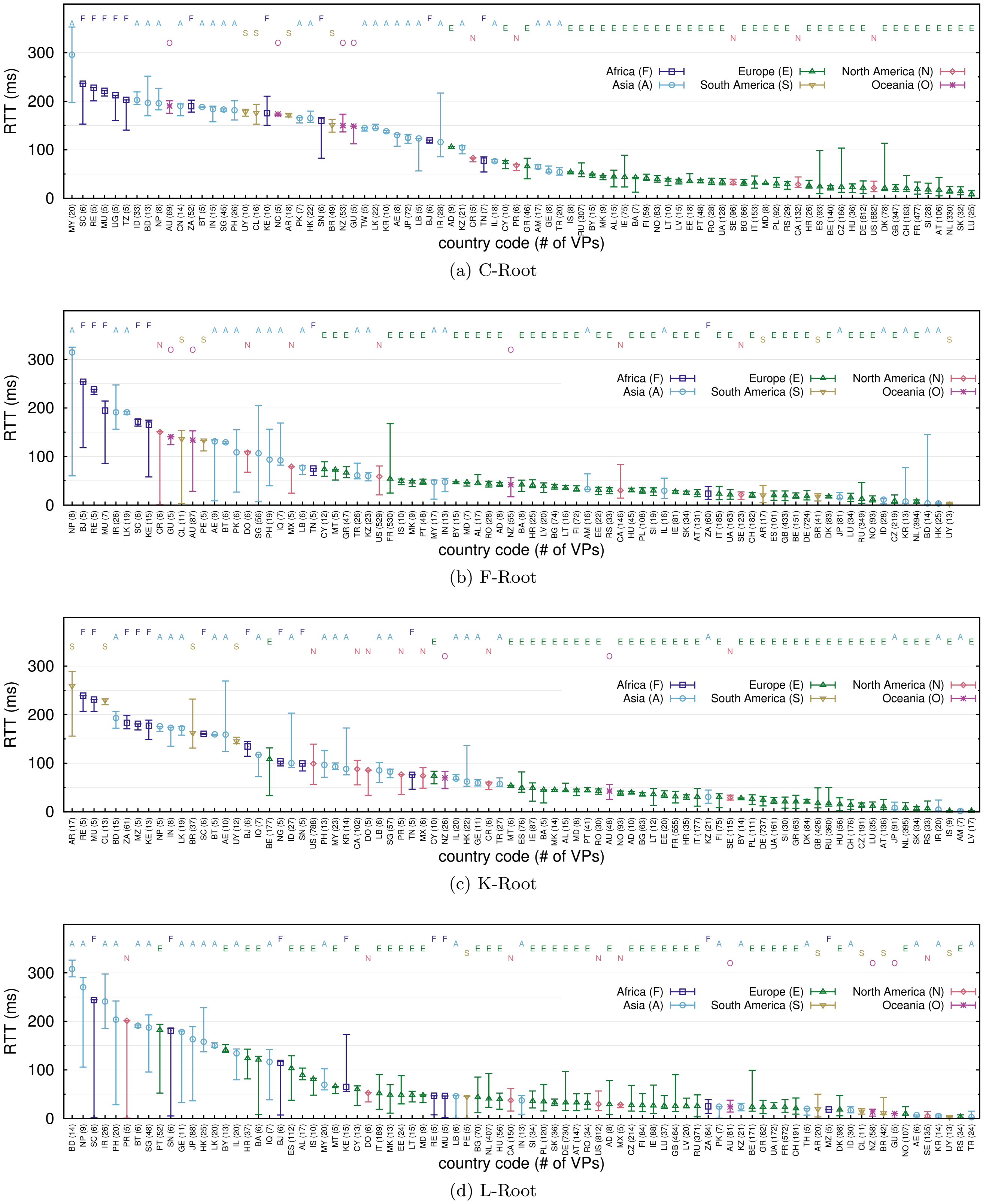
[39] Spring, N., Mahajan, R., and Anderson, T. Quantifying the causes of path inflation. In *Proceedings of the ACM SIGCOMM Conference* (Karlsruhe, Germany, Aug. 2003), ACM, pp. 113–124.

[40] Streibelt, F., Böttger, J., Chatzis, N., Smaragdakis, G., and Feldman, A. Exploring EDNS-Client-Subnet Adopters in your Free Time. In *Proceedings of the ACM Internet Measurement Conference* (2013), IMC, pp. 305–312.

[41] Toonk, A. How OpenDNS achieves high availability with anycast routing. https://labs.opendns.com/, 2013.

[42] Torres, R., Finamore, A., Kim, J. R., Mellia, M., Munafò, M. M., and Rao, S. Dissecting Video Server Selection Strategies in the YouTube CDN. In *Proceedings of the 31st International Conference on Distributed Computing Systems* (2011), ICDCS, pp. 248–257.

[43] Woolf, S., and Conrad, D. Requirements for a Mechanism Identifying a Name Server Instance. RFC 4892, 2007.



(a) C-Root

(b) F-Root

(c) K-Root

(d) L-Root

Figure 12: Median RTT (quartiles as error bars) for countries with at least 5 VPs (number of VPs for each country is given between parenthesis). Letters at top indicate continents.