Broad and Load-Aware Anycast Mapping with Verfploeter

Wouter B. de Vries
University of Twente

Ricardo de O. Schmidt
University of Twente

Wes Hardaker
USC/ISI

John Heidemann
USC/ISI

Pieter-Tjerk de Boer
University of Twente

Aiko Pras
University of Twente

ABSTRACT

IP anycast provides DNS operators and CDNs with automatic failover and reduced latency by breaking the Internet into *catchments*, each served by a different anycast site. Unfortunately, *understanding* and *predicting* changes to catchments as anycast sites are added or removed has been challenging. Current tools such as RIPE Atlas or commercial equivalents map from thousands of vantage points (VPs), but their coverage can be inconsistent around the globe. This paper proposes *Verfploeter*, a new method that maps anycast catchments using active probing. Verfploeter provides around 3.8M passive VPs, 430× the 9k physical VPs in RIPE Atlas, providing coverage of the vast majority of networks around the globe. We then add load information from prior service logs to provide calibrated predictions of anycast changes. Verfploeter has been used to evaluate the new anycast deployment for B-Root, and we also report its use of a nine-site anycast testbed. We show that the greater coverage made possible by Verfploeter's active probing is necessary to see routing differences in regions that have sparse coverage from RIPE Atlas, like South America and China.

CCS CONCEPTS

• **Networks** → **Network design principles**; **Network design and planning algorithms**; **Network measurement**; **Naming and addressing**; *Network layer protocols*; *Network resources allocation*; *Network performance analysis*; *Denial-of-service attacks*; *Logical / virtual topologies*; *Overlay and other logical network structures*;

KEYWORDS

DNS, IP anycast, catchments, Internet mapping, service provisioning

ACM Reference Format:
Wouter B. de Vries, Ricardo de O. Schmidt, Wes Hardaker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. 2017. Broad and Load-Aware Anycast Mapping with Verfploeter. In *Proceedings of IMC '17, London, United Kingdom, November 1–3, 2017,* 12 pages.
https://doi.org/10.1145/3131365.3131371

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IMC '17, November 1–3, 2017, London, United Kingdom
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5118-8/17/11…$15.00
https://doi.org/10.1145/3131365.3131371

1 INTRODUCTION

IP anycast allows an Internet service operator to provide services such as DNS or HTTP content delivery from multiple sites that are, usually, physically distributed [2]. Anycast can reduce latency [28, 43] and blunt Distributed Denial-of-Service (DDoS) attacks by spreading traffic across different sites and providing greater aggregate capacity than any one site might [33]. IP anycast is used by Root DNS operators [2, 41], commercial DNS service providers and operators of top- and second-level domains. It is also used by multiple Content Distribution Networks (CDNs), including Microsoft/Bing [19], Verizon/Edgecast, and others.

Anycast operates by deploying servers at different *sites*[1], and having each site announce the same IPv4 or v6 prefix using the Border Gateway Protocol (BGP—the standard protocol for inter-AS routing). All networks that receive these routes from BGP select their topologically closest site, as defined by the BGP metrics and policies. These networks define the *catchment* of that anycast site. User send queries that are routed to the nearest anycast site; because routing is pre-computed, there is no query-time cost for this site-selection.

Understanding anycast catchments is important for performance (throughput, latency, and load balancing), defending against DDoS, and managing filtering. Anycast operators engineer their deployments to minimize latency to users, and to spread load over multiple sites. Some anycast systems employ tens or hundreds of anycast sites to minimize latency to users [10, 43], although evaluating how well that goal has been achieved requires ongoing observation [9, 43]. Operators also often balance load across multiple sites to manage capacity [11, 19], particularly for CDNs providing large volumes of content. Load balancing across anycast sites is particularly important to mitigate DDoS attacks [33], where matching attack traffic to capacity or isolating attack traffic to certain catchments are essential tools.

In addition to performance, anycast catchments can interact with country-specific policies for content filtering. Filtering of DNS to implement national-specific policies is not uncommon [3, 21, 23, 50], so it is important that policies and catchments align. Two examples of Root DNS service show cases where there is a mismatch. In 2010, the catchment I-Root DNS service's site in Beijing expanded outside China, imposing China's censorship policies outside its borders [30]. More recently, at the beginning of 2017, the catchment for a K-Root anycast site in Iran was seen outside that country, inconsistent with the policies of the K-Root operators and the hosts of that site [1, 29].

[1] Anycast documents sometimes use the term *instance*, but that term can apply to both sites or individual servers. We avoid the term "instance" because it is ambiguous when anycast sites have multiple servers, as is often the case. Similarly, RFC 4786 [2] uses the term *node* for what we call a site; we avoid node because it often refers to specific servers.

The challenge in managing anycast is that BGP routing is not always what one would expect. Absent other policies, BGP defines nearness in terms of AS-hops, but one AS hop across an organization with a global network (such as AT&T, Tata, and NTT) can have very different latency than one AS hop across a small ISP. In addition, the trend of a flatter Internet [27] means that AS hops provide coarser control than it did in the past. More importantly, BGP policy controls allow ISPs to manipulate routing for business reasons; policy controls are widely used to do traffic management.

Current approaches to manage anycast catchments use one-off active measurements [11], platforms for active measurement such as RIPE Atlas [12, 43], commercial services (for example, [44]), and analysis of anycast service logs [22]. While these approaches have provided insight, and RIPE Atlas [40] and commercial services are in wide use, even the largest services have relatively small numbers of vantage points (from hundreds to 10,000 or so), and it is unclear how these measurement systems relate to actual operational traffic. Analysis of anycast service logs offer an accurate representation of actual load, but require the anycast service to be in operation and active use.

The overall contribution of this paper is to provide a new approach to mapping anycast catchments (§3) that has been validated through real world ground truth. This approach provides *broad coverage* and can be combined with traffic history to provide *estimated load*, providing *operational value* for one anycast service, and an approach that can be used by others. The insight in our new measurement approach is to use *active probing using the anycast service itself* and we can use *historical traffic to predict future load*. In §5 we show that active probing allows coverage of the ping-responsive Internet, currently about 4M /24 networks, providing 430× more information than current public measurement platforms. By contrast, coverage from existing platforms scale relative to the ability to deploy physical devices or virtual machines, both of which are limited.

The second contribution of this work is to use Verfploeter to examining the operational catchment for B-Root and to study anycast in Tangled, a nine-site anycast testbed (§6). B-Root deployed anycast only recently (May 2017), and *our approach contributed to the success of this planning and deployment*. Analyzing this active network deployment allows us to compare the predictive capability of our approach to prior approaches such as RIPE Atlas. Evaluation of our Tangled testbed lets us test a larger anycast deployment (nine sites compared to B-Root's two sites). Our approach provides a new way to evaluate anycast stability with much broader coverage than recent studies [48].

Although our case study with B-Root and Tangled focus on DNS, Verfploeter can examine any anycast service, although load prediction requires a system that can estimate historical traffic load.

A complete version of Verfploeter is available as open source at https://github.com/woutifier and https://ant.isi.edu/software/lander/. We have released all the data used in this paper (except *LN-4-12*, which is not ours); see citations in Table 1 and Table 2.

2 RELATED WORK

There have been several prior approaches to measure anycast catchment using a variety of techniques.

Use of Open Resolvers: Early work used Open DNS Resolvers in combination with PlanetLab and Netalyzr to map catchments of anycast services [18]. While Open Resolvers provided a broad view at the time of their study (300k VPs), they are being steadily shut down out of concerns about their use in DNS amplification attacks [31]. While open resolvers offered a very large set of vantage points, they are fewer than the method we propose that uses ping-responsive networks. (A direct comparison is potential future work.)

Measurement Platforms: The most common method of assessing anycast is to use public or private measurement platforms that offer physical or passive VPs around the Internet. RIPE Atlas [40] and PlanetLab [34] are both openly available and widely distributed, and a number of commercial platforms are also available. Systems we are aware of range from hundreds to around 10k VPs.

Several studies, both by others and us, have used measurement platforms to study anycast [1, 9, 11, 12, 18, 30, 33, 43]. As pre-deployed measurement platforms these systems are available and can measure anycast services externally (without requiring support from the service operator). The main weaknesses of these systems are that they are slow and expensive to grow, and deployment is often skewed relative to the population of Internet users. This skew has been noted in many prior studies and was recently studied explicitly [8].

Traffic and Log Analysis: Anycast operators have always been able to assess current anycast performance by analyzing their own traffic and server logs. Recent work examined the Microsoft Bing CDN [11] and a variety of other CDNs [22]. As the service operator, log analysis requires no external measurements and can cover the entire service. While important, analysis of existing services can only study the *current* deployment—it requires active use by a large number of users and cannot directly support pre-deployment planning. Second, log files may be unavailable due to privacy concerns, cost of storage or retrieval, or concerns about performance impact on operational services. We use logs when available, but do not require them.

Performance Analysis of DNS Services: There have been a number of analysis of root DNS service, both pre-anycast [20] and with anycast for latency [9, 13, 18, 28, 43] and DDoS [33].

To the best of our knowledge, our paper is the first to present this ICMP-based anycast catchment determination approach. Further, we do not know of any larger scale catchment measurement with open datasets against a real-world anycast deployment.

3 VERFPLOETER: GLOBAL PROBING

Our approach has components to map anycast catchments for a large fraction of prefixes in the Internet, and to estimate load from each of these prefixes.

3.1 Mapping Anycast Catchments

Traditional approaches to measuring anycast catchments use many vantage points (VPs) around the Internet; each VP queries the anycast service to determine its catchment. In prior work for DNS, the VPs are typically RIPE Atlas probes [39, 43], and the queries use DNS TXT records, with the special CHAOS network type, and the name "hostname.bind" [49], or the newer NSID option in DNS [4]. One can augment these methods with traceroutes to detect possibly

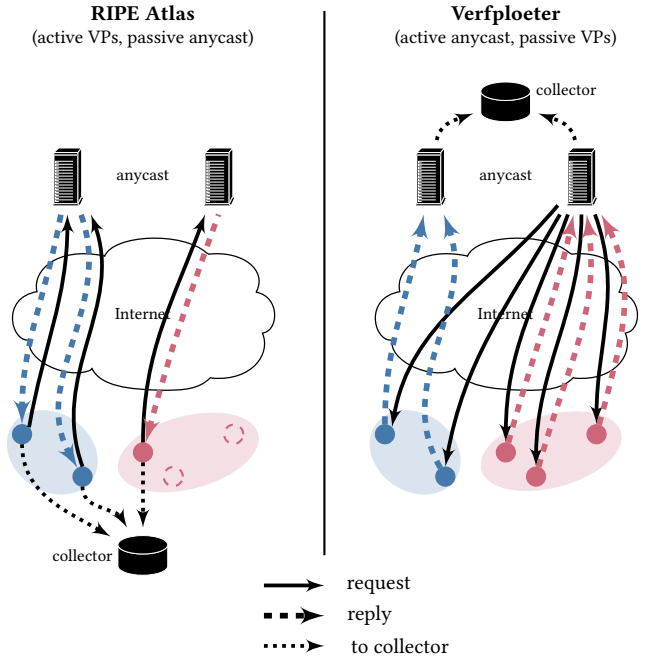


Figure 1: Traditional catchment mapping from active VPs using a testbed like RIPE Atlas (left); and using Verfploeter with queries originating in the anycast system (right).

spoofed replies [18]. All of these approaches require deployment of active probes around the Internet. The largest studies we know of use between 9000 and 10000 VPs, all the active VPs in RIPE Atlas.

Our insight is that we do not need control over active vantage points if we can *solicit* messages from around Internet that will identify their catchment. Rather than handle both queries and responses from the VPs, we instead generate queries that cause *VPs to respond and reply to the anycast system*; we define these as **passive VPs**. If we can capture traffic at all anycast catchments, we can determine the catchment of each VP that responds. In effect, we shift the active side that generates and receives queries from the VP to the anycast network itself, yet capture observations from millions of passive VPs. (Although the anycast sites capture the data, the ping targets are the vantage points because they each generate a catchment report.)

Figure 1 compares these methods. On the left, traditional mapping sends queries (black arrows) from VPs into the anycast system. On the right, we send queries *from* the anycast network block (defined by the source address), *to passive VPs* in most /24 IPv4 networks. Their replies return to the site for their catchment, even if it is not the site that originated the query.

In Verfploeter, our *queries* are ICMP Echo Requests (pings), sent using a custom program, soliciting ICMP Echo Replies. Queries are sent from a designated measurement address that *must be in the anycast service IP prefix*. Unlike traditional catchment mapping, it is not the reply payload that indicates catchment, but instead *the catchment is identified by the anycast site that receives the reply.*

Our *passive VPs* are any computers in the Internet that reply to pings. We use a recent ISI IPv4 hitlist [17]. In principle, we could ping every IPv4 address to get complete coverage from all addresses that reply. We use hitlists instead because they provide representative addresses for each /24 block that are most likely to reply to pings, and with one address per /24 block, we can reduce measurement traffic to 0.4% of a complete IPv4 scan. (We select /24s as the smallest routable prefix in BGP today, since anycast depends on BGP.)

We send requests in a pseudorandom order (following [25]), and relatively slowly (about 6k queries per second), to spread traffic, limiting traffic to any given network to avoid rate limits and abuse complaints. Although well known techniques would allow much faster probing, there is little penalty for probing over 10 or 20 minutes.

We must capture traffic for the measurement address with our response collection system. We can capture traffic at the routers (without having a computer at the address), or by running computers that capture traffic on the address itself. These captures must happen concurrently at *all anycast sites*. We have three different *response collection systems*: first is a custom program that does packet capture and forwards responses to a central site in near-real-time. Second, we collect replies with LANDER [26], an existing packet capture system that collects data continuously. Third, we have also used tcpdump directly to capture traffic specifically for the measurement address. We use the first method for Tangled and both of the second methods at B-Root.

Capturing traffic at sites may be a requirement for some anycast operators, but not a large one. While the measurement address is in the service /24, it can be a different address and need not see non-measurement traffic. Operators already operate services on these networks, and measurement can be done either on a virtual IP address associated with the computer providing service, on dedicated measurement hardware, or by using virtual machines on the same network. Measurement should be time synchronized across all sites so they can be easily combined, but standard techniques like NTP are sufficient.

We send only a single request per destination IP address, with no immediate retransmissions. We see replies from about 55% of blocks (Table 4), consistent with the 56% and 59% seen in previous studies [17]. While incomplete, we get responses for millions of blocks. We could improve the response rate by probing multiple targets in each block (as Trinocular does [36]), or retrying immediately. Exploration of these options is future work. Finally, we copy all responses to a central site for analysis. Total traffic across the service is about 128 MB per measurement, so it is not huge. We currently copy data manually, or with a custom program that forwards traffic after tagging it with its site.

Our approach to catchment mapping requires active participation at all anycast sites—it requires cooperation of the anycast operator, but it does not require additional Internet-wide infrastructure (such as distributed VPs). Fortunately, anycast operators are strongly motivated to understand their systems. These trade-offs are the opposite of traditional anycast mapping, which requires active VPs but not support of the target anycast system.

We do not model BGP routing to predict future catchments, *we measure actual deployment*. To predict possible future catchments

from different policies, one must deploy and announce a test prefix that parallels the anycast service, then measure its routes and catchments. (We assume the test prefix will encounter the same policies as the production prefix.) Fortunately, anycast providers often announce anycast on a /24 prefix, and a larger, covering, /23 prefix with unicast (this approach protects against corner cases with some routing policies [14]). The non-operational portion of the /23 could serve as the test prefix.

3.2 Load Estimation

Planning anycast deployment is more than just mapping catchments—different services can experience very different loads, depending on the distribution and usage patterns of its client base. We therefore build load estimates for each network block (/24 prefix) that accesses a service, so we can calibrate the loads that will be generated by a given catchment.

We assume operators collect query logs for their systems and can use the recorded historical data to estimate future loads. (For example, all root operators collect this information as part of standard RSSAC-002 performance reporting [42].) For our study of B-Root we use historical data from its unicast deployment. When no operational load data is available, as in Tangled, one must estimate load using data from a similar service, or assume uniform load if no better estimates are available.

We consider three types of load: queries, good replies, and all replies. Queries represent incoming load on the servers, while replies are the results. Query packet load counts may differ from replies if response rate limiting is used to blunt DNS amplification attacks [45]. We separate out good replies from all replies because of the large fraction of queries to non-present domains in root-server traffic (first observed in 1992 [15] and still true today); operators may wish to optimize for volume or for good replies.

In principle, we can estimate load over any time period. Practically, we compute it over one day, and look at overall traffic using hourly bins.

4 MEASUREMENT SETUP AND DATASET

Using the proposed ICMP-based method, Verfploeter, we measure the catchment of two anycast services, B-root and an anycast testbed (Tangled), from more than 6.4M VPs (IP addresses). (Table 1 lists all datasets we use, and each figure or table reports the dataset it uses in the caption.) We add geolocation information for these blocks using MaxMind [32]. Accuracy of this geolocation is considered reasonable at the country level [35]. We also use Route Views and RIPE RIS data to determine the AS number for each scanned IP address and the prefixes that are announced by each AS.

Data cleaning: We remove from our dataset the duplicate results, replies from IP-addresses that we did not send a request to, and late replies (15 minutes after the start of the measurement). Duplicates are caused by systems replying multiple times to a single echo request, in some cases up to thousands of times, accounting for approximately 2% of all replies. Other systems, when pinged, reply from a different IP-address than the original target destination. Methods such as alias resolution might clarify this, however, further investigation is out of the scope in this paper.

Id	Service	Method	Start	Dur.
SBA-4-20	B-Root	Atlas	2017-04-20	8 m
SBA-4-21	[38]		2017-04-21	8 m
SBA-5-15			2017-05-15	10 m
SBV-4-21	B-Root	Verf-	2017-04-21	20 m
SBV-5-15	[24]	ploeter	2017-05-15	20 m
STA-2-01	Tangled [46]	Atlas	2017-02-01	10 m
STV-2-01	Tangled	Verf-	2017-02-01	10 m
STV-3-23	[47]	ploeter	2017-03-23	24 h

Table 1: Scans of anycast catchments for B-Root and our testbed (Tangled). Scans were done on various days for comparison. Dataset *STV-3-23* contains 96 measurements over 24 hours, each 10 minutes long.

				Queries	
Id	Service	Date	Site	q/day	q/s
LB-4-12	B-Root [6]	2017-04-12	LAX	2.34G	27.1k
LB-5-15	B-Root [7]	2017-05-15	both	2.20G	25.4k
			LAX	1.78G	20.6k
			MIA	0.407G	4.71k
LN-4-12	NL ccTLD	2017-04-12		redacted	

Table 2: Datasets used to study load (IPv4 UDP queries only).

4.1 B-Root

We validate the proposed methodology by providing a detailed view of the catchment of one of the DNS root-servers. B-root is the most recent root letter to make the change from unicast to anycast. B-Root deployed anycast at the beginning of May, 2017 [5], adding a site in Miami to its original site in Los Angeles (Table 3).

B's new deployment of anycast makes it an interesting analysis target. Unlike the other DNS Roots, B does not have a history of anycast deployment to guide its choices (although of course it draws on experience of other anycast deployments).

Dataset: We study B-Root catchments using several scans using both RIPE Atlas and Verfploeter, as shown in Table 1. We estimate B-Root load using two day-long datasets listed in Table 2. As a baseline we use data from DITL 2017 (A Day in the Life of the Internet [16]), taken Wednesday, 2017-04-12 (UTC), before B-Root was using anycast. We then test against Thursday, 2017-05-15 (UTC), after B-Root anycast was well established.

4.2 Anycast Testbed

We augment our measurements of B-Root with measurements of our anycast testbed, *Tangled*. This testbed has 9 sites around the world: 5 sites in Europe, 2 in the USA, and 3 other sites spread across Asia, Oceania and South America (Table 3). Tangled allows us to study how a larger amount of sites interact, and to perform experiments which we cannot do in an operational anycast service. We use it to understand anycast instability and ASes that appear in multiple catchments (§6).

Limitations: Three of the testbed sites share a common ISP, which might impact the overall catchment. The anycast site in São Paulo has all its traffic routed via the same link as the site in Miami, which might cause announcements from São Paulo to be

Service	Location	Host	Upstream
B-Root	US, Los Angeles	USC/ISI	AS226
	US, Miami	FIU/AMPATH	AS20080
Tangled	AU, Sydney	Vultr	AS20473
	FR, Paris	Vultr	AS20473
	JP, Tokyo	WIDE	AS2500
	NL, Enschede	Univ. of Twente	AS1103
	UK, London	Vultr	AS20473
	US, Miami	Florida Int. Univ.	AS20080
	US, Washington	USC/ISI	AS1972
	BR, Sao Paulo	Florida Int. Univ.	AS1251
	DK, Copenhagen	DK Hostmaster	AS39839

Table 3: List of anycast sites used in our measurements.

hidden. Finally, the connectivity at the site in Japan is such that it does not attract much traffic since announcements from other sites are almost always preferred over it. Prior to the measurement, the connectivity of each site was validated individually by announcing our prefix from that location only. Such limitations are not particular to our testbed as similar features can also be observed in public anycast services [43].

Dataset: As shown in Table 1, we measured the catchment using both Verfploeter and Atlas on Wednesday, 2017-02-01 (UTC). We also determined the catchment of Tangled, using only Verfploeter, every 15 minutes during a 24 hour period starting 2017-03-23 10:57 UTC, for a total of 96 measurements. In total we collected 342,604,759 ICMP replies, of which 324,675,876 remained after cleaning.

For each measurement we transmitted one ICMP packet to each of the 6.4M IPs from the hitlist, at a rate of 10k/second to prevent overloading networks or network equipment. Each measurement round took approximately 10.5 minutes to complete. A unique identifier in the ICMP header was used in every measurement round to ensure data set separation.

5 ANALYSIS OF THE VERFPLOETER MECHANISM

In this section we examine the Verfploeter measurement method. We show the broader coverage of Verfploeter compared to RIPE Atlas, and how catchment mapping from Verfploeter can be combined to historic traffic load to accurately predict load at individual anycast sites.

5.1 Utility: Operational Evaluation of Anycast Catchments

A long-standing goal of anycast mapping is to assess load balancing and routing problems [9, 43]. We next look at B-Root's anycast distribution. Deployed recently in May 2017, it has only two sites, but we are able to deploy Verfploeter on it.

We have measured the geographic footprint of B-Root with RIPE Atlas (Figure 2a) and Verfploeter (Figure 2b). These maps highlight a couple of important differences between these measurement methods.

First, Verfploeter *has much broader coverage*: Atlas coverage is good in Europe and reasonable in North America, but sparse elsewhere and almost absent in China. Verfploeter provides good coverage for most of the populated globe. Second, even where coverage is good, Verfploeter *provides far more numerous observations*—the scale of Figure 2b is 1000× greater than Figure 2a.

These differences are particularly important for examination of B-Root catchments in South America and China. The broader coverage is important to understand, for example, how a host in China might select a B-Root site: Atlas cannot comment, but Verfploeter shows most of China selects the MIA site.

The denser coverage in South America also helps highlighting the impact of B-Root's hosting ISPs. B-Root's ISP in MIA (AMPATH) is very well connected in Brazil and Argentina, but does not have direct ties to the west coast of South America. This difference shows in the wider use of the MIA site in Brazil, and less use of it in Peru and Chile.

Better coverage in locations like these that currently have poorer coverage by RIPE Atlas are important, particularly since East and South Asia are home to many Internet users but few Atlas VPs.

B-Root's goal in measuring anycast is to understand routing choices; we return to this question in §6.1.

5.2 Utility in Mapping Multi-Site Anycast

B-Root shows the benefits of increased number of VPs with Verfploeter, but we would like to understand how the different approaches work on anycast deployments with more sites. We therefore turn to Tangled: an anycast testbed designed and deployed by us (§4.2).

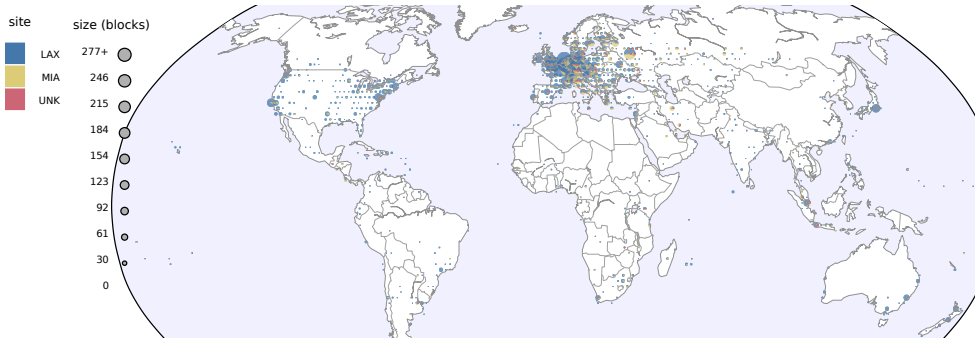
Figure 3 maps the catchments of Tangled with Atlas and Verfploeter. Again, outside of Europe, the greater density of coverage of Verfploeter provides clear qualitative differences between the two maps. For example, the IAD site (dark yellow) shows up prominently across North America with Verfploeter, but with Atlas, CDG and ENS seem to serve that region. We also see very different mixes of sites in Australia. And only Verfploeter provides coverage of China.

The key result from these graphs is that *Verfploeter coverage tracks the Internet as a whole*, not just where physical VPs can be placed. We quantify this difference in the next section.

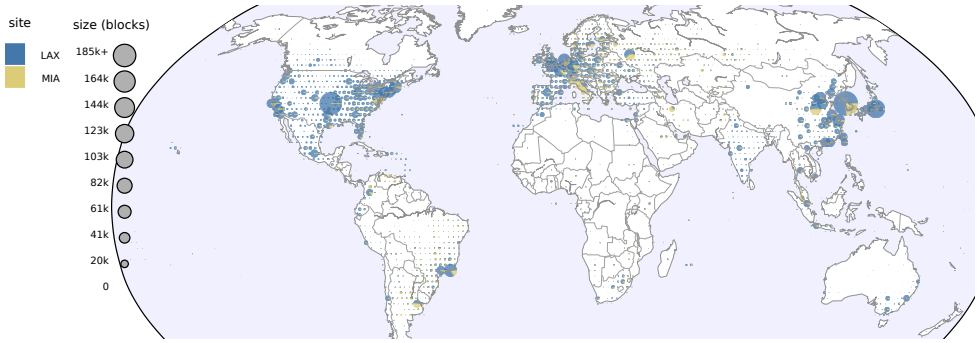
5.3 Greater Coverage in Verfploeter

In §5.1 and §5.2 we showed how the greater coverage in Verfploeter reveals aspects of B-Root and our testbed Tangled that would otherwise be missed. This coverage is possible because Verfploeter's passive VPs only require a computer that responds to ICMP, instead of physically deployed devices (Figure 1); this way we can cover millions of /24s.

To quantify the difference in coverage that is visible in Figure 2, Table 4 compares how many blocks the two measurement approaches see. For both systems we try to use all available VPs, but some VPs are unavailable: for Atlas, 455 VPs do not respond (within 406 blocks), presumably because they are temporarily down. For Verfploeter, about 3M ping targets do not reply, presumably because the target was temporarily down, or it was in a block of dynamic addresses and temporarily unused. If desired, both of these



(a) RIPE Atlas coverage of B-Root (Dataset: *SBA-5-15*)



(b) Verfploeter coverage of B-Root (Dataset: *SBV-5-15*).

Figure 2: Geographic coverage of vantage points for RIPE Atlas and Verfploeter for B-Root, in two-degree geographic bins. The pie in each bin is colored by site (blue: LAX; yellow: MIA; red: other). Circle areas show number of address blocks (Verfploeter) or VPs (Atlas) at different scales.

	RIPE Atlas (VPs)	RIPE Atlas (/24s)	Verfploeter (/24s)
considered	9807	9083	6,877,175
non-responding	455	406	3,090,268
responding	9352	8677	3,786,907
no location	0	0	678
geolocatable	9352	8677	3,786,229
unique		2079	3606300

Table 4: Coverage of B-Root from the perspective of the RIPE Atlas and Verfploeter measurement systems, measured in VPs (Atlas) or /24 blocks (both). (Datasets: *SBA-5-15*, *SBV-5-15*)

non-response rates could be reduced by retrying later, or with additional addresses for Verfploeter. All Atlas VPs have geolocation (set when the VP is registered), but we discard a few Verfploeter blocks (678) that we cannot geolocate.

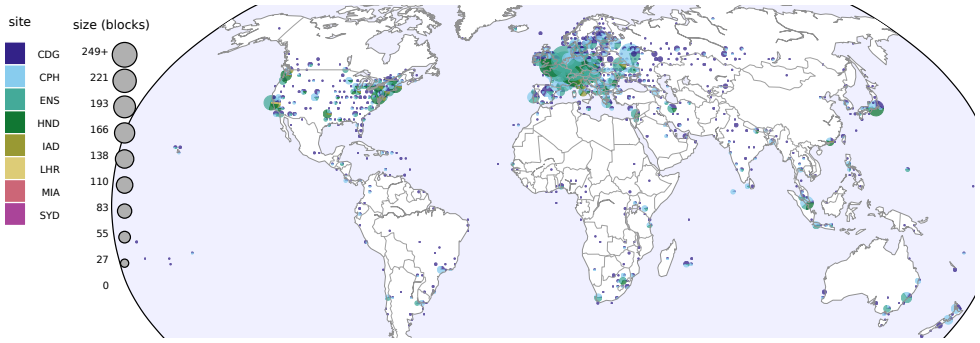
The key result about coverage is that Verfploeter *sees around* 430× *more blocks than Atlas*. Although Atlas finds a few unique blocks (presumably blocks that discard all pings), about 77% of Atlas blocks are also seen by Verfploeter, and Verfploeter sees around 3.61M additional blocks.

5.4 From Observations to Load

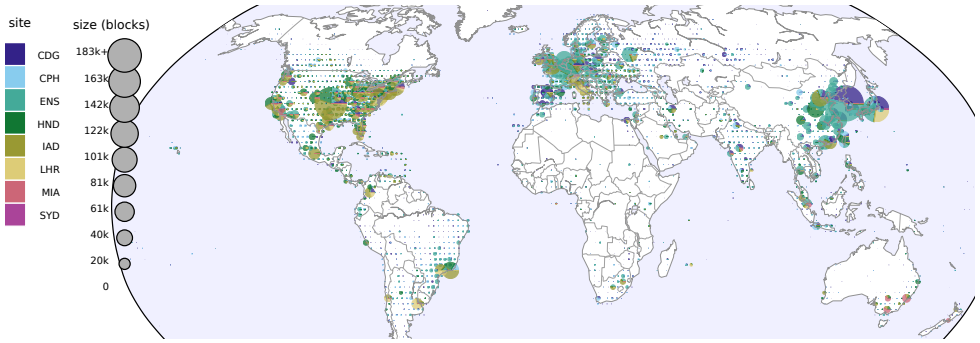
We next look at how well different measurement systems relate to actual load on an anycast service. It is well known that the distribution of RIPE Atlas reflects more about who RIPE interacts with than global Internet traffic—as an European project, and Europe being the main region of RIPE NCC operation, Atlas' deployment is by far heavier in Europe than in other parts of the globe (and this is a well known shortcoming [8]). Our goal here is to *calibrate* different measurement systems to best match actual traffic. We show that, once calibrated, we can get very accurate predictions about expected service load, but the calibration is necessary to account for variation in load per block. Calibrated predictions are important if Verfploeter is to be used for capacity planning.

Estimating Load: To estimate load on B-Root, we begin with our prediction about anycast catchments from Verfploeter, then we weight each /24 block by our measurements of its known traffic load (§3.2). There are blocks for which we do not have anycast mapping, either because they do not reply to our probes, or because the specific address we chose to contact did not reply; these blocks are mapped to "unknown", indicating we cannot determine the anycast mapping. (Although we assume their traffic will go to our sites in similar proportion to blocks in known catchments.)

Figure 4a shows the result of this load prediction. It is useful to compare this estimate to Figure 2b, which counts /24 blocks that source traffic, and Figure 2a, which counts Atlas VPs.



(a) RIPE Atlas coverage of Tangled (Dataset: *STA-2-01*).



(b) Verfploeter coverage of our nine-site testbed (Dataset: *STV-2-01*).

Figure 3: Catchments for Tangled from RIPE Atlas and Verfploeter. Circle areas show number of blocks (Verfploeter) or VPs (Atlas) at different scales; each is a pie chart with colors showing each site.

	Blocks		Queries	
	/24s	%	q/day	%
seen at B-Root	1,388,338	100%	2.19G	100%
mapped by Verfploeter	986,605	87.1%	1.80G	82.4%
not mappable	401,733	12.9%	384M	17.6%

Table 5: Coverage of Verfploeter from B-Root. (Dataset: *SBV-5-15, LB-5-15.*)

Date	Method	Measurement	% LAX
2017-04-21	Atlas	967 VPs	68.8%
2017-05-15		9,682	82.4%
2017-04-21	Verf-	4.069M /24s	82.4%
2017-05-15	ploeter	3.923M	87.8%
2017-05-15	+ *load*	n/a q/day	*81.6%*
2017-05-15	*Act. Load*	*2.188G* q/day	*81.4%*

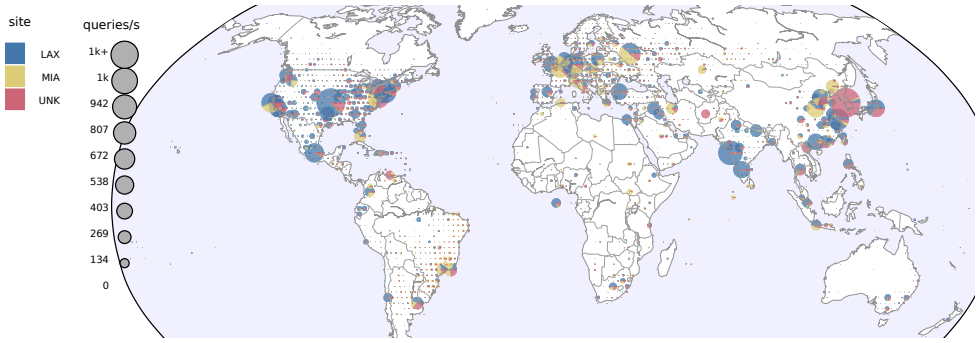
Table 6: Quantifying differences B-Root anycast with different measurement methods and times.

The most striking operational difference between measurements of blocks and actual load estimates is that load seems to *concentrate* traffic in fewer hotspots. This outcome should not be surprising: DNS is a common service operated by most ISPs with a local recursive resolver. Thus an ISP with users spread over a large region may still send all DNS traffic through recursive resolvers housed at a few data centers. Weighting coverage by load corrects for these protocol-specific effects that are not seen directly in our ICMP-based measurements.

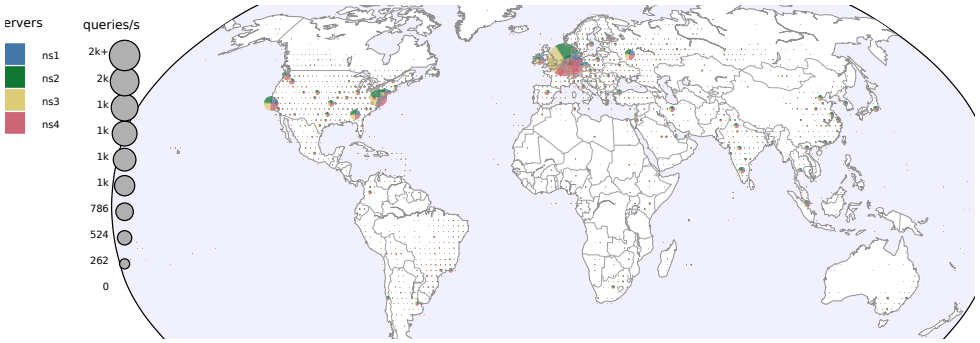
Second, Verfploeter can only map blocks that respond to our probes. Table 5 shows coverage as seen from B-Root's traffic logs, showing that there are a large number of blocks (about 12.9%) that are not mapped. Figure 4a plots the load from these blocks in red, showing that most are in Korea, with some in Japan and central and southeast Asia. In §5.5 we show that these missing blocks do not alter our predictions.

Finally, we see that load is higher in some regions than the number of blocks would suggest, particularly in India. This difference may be explained by many users using relatively few IP blocks in these areas, with a great deal of deployed network address translation behind those blocks.

Quantifying Differences from VPs to Blocks to Load: While Figure 2 and Figure 2b show visual differences, we turn to Table 6 to quantify those differences and their impact on assessment of catchment sizes in B-Root. When we compare Atlas, Verfploeter, and Verfploeter with load, we see very different measurements (thousands of VPs, millions of blocks, or billions of queries per day). Load estimates (§3.2) determine different weighting factors and result in different fractions of traffic between the LAX and MIA sites, as shown in the "% LAX" column. In §5.5 we will compare these values to measured load to see which is most accurate, but



(a) Geographic distribution of load by site for B-Root, as inferred from Verfploeter (Datasets: *LB-4-12*).



(b) Geographic distribution of load for .nl, as determined by traffic logs (Dataset: *LN-4-12*).

Figure 4: Measured DNS traffic over geography for B-Root and .nl.

next we see how these changes will be even larger for DNS services with less even global load.

Uneven Load: Load for B-Root is global and largely follows the distribution of Internet users, so Figure 4a has only moderate differences from Figure 2b.

Other DNS systems are more regional. Figure 4b shows load for four of the .nl nameservers, the country domain for the Netherlands. They cannot easily collect data from their two nameservers that use anycast, so they are omitted from this plot and it may underrepresent global traffic, but we know it captures at least half of all global traffic to this domain.

Unlike B-Root, we see that the majority of traffic to .nl is from Europe and the Netherlands. There is also significant traffic from the U.S. and some global traffic. With this type of client distribution, calibrating the measured catchment using load information is critical.

5.5 Using Verfploeter to Predict Load

We next examine how accurate Verfploeter's load modeling can *predict future load*. Our goal is to determine how much unmappable blocks (§5.4) affect accuracy, and how much routing and load shifts over time. In both cases we observe partial information and predict load for the unobserved remainder (observing responses per blocks and predicting load, or observing load now and predicting future load), then compare that against complete information. A study of long-term predictions will require more experience with Verfploeter, but we address the basic accuracy question here.

We study the accuracy of load predictions with Verfploeter by analyzing what network blocks B-Root sees traffic from that Verfploeter has found to be unmappable by examining the DNS network load at B-Root on 2017-05-15 (Dataset: *LB-5-15*) and the Verfploeter analysis performed on the same day (Dataset: *SBV-5-15*). (Since Tangled is not a production service, we cannot study its operational load.) Recall from Table 6 that although Verfploeter finds 87.8% of network blocks reach LAX, the load prediction is that 81.6% of traffic should go to LAX. That prediction does not consider blocks that send traffic to B-Root but do not respond to Verfploeter (12.9% from Table 5).

Predicted vs. Measured Load: The last line of Table 6 shows the *actual* load of 81.4%, as measured at all B-Root sites on 2017-05-15. We see our 81.6% prediction using same-day Verfploeter and load is quite close to the measured result. Our first observation is that this result suggests *Verfploeter-unobservable blocks do* not *have significant effects* on our overall load estimate. (Future work could strengthen this claim by demonstrating it for services other than B-Root.) Although they account for 17.6% of queries (Table 5, and the red slices in Figure 4a), the fraction of traffic that goes to each B-Root site appears to follow the ratio seen in measured blocks.

Our second observation is that *our load-weighted predictions are very close to observed load.* Verfploeter without load adjustment is further off, with 87.8% of blocks going to LAX. We conclude that weighting by load is important. Surprisingly, Atlas estimates, at 82.4%, are actually closer than Verfploeter if Verfploeter is not load-weighted.

The key take-away of this result is that with load-weighted Verfploeter *preliminary results suggest it is possible to make reasonable predictions about future anycast deployments* by measuring the deployment on a test network and predicting future traffic levels using recent load data. We hope to expand these results beyond B-Root as ongoing work.

Long-duration predictions: Finally, we can also look at long-duration prediction. We performed a similar prediction analysis in advance of the B-Root deployment using the Verfploeter data gathered on 2017-04-21 and network traffic from 2017-04-12. We see a fairly large shift in blocks between these dates, with Verfploeter shifting from 82.4% to LAX in April to 87.8% in May. By weighting the *SBV-4-21*Verfploeter dataset from the B-Root test prefix with the *LB-4-12* measured load, we find that the predicted DNS request load arriving at LAX is 76.2%. This is significantly less than the 81.6% measured load in *LB-5-15*, which highlights the discrepancy between shifts in routing over one month between the *SBV-4-21* and *SBV-5-15* dataset collection periods.

This shift suggests that the accuracy of load estimates depends on how old the data is. We know that routing changes in the Internet over time [9]; this early result suggests some care must be taking with long-duration predictions. We expect that predictions further into the future will be less accurate than short-term predictions. While we are collecting data to answer this question, such a study is future work.

6 RESULTS: UNDERSTANDING ANYCAST WITH VERFPLOETER

We next use Verfploeter to explore three questions about anycast. These questions have each been raised in prior work; here we use Verfploeter to revisit them (and compare to them, in §6.1 and §6.3), both to show its utility and to refine these prior results.

6.1 Use of AS Prepending in B-Root

An important operational question for B-Root is understanding how to balance load between sites. Although both sites are able to handle normal traffic, DNS operators need to shift load during emergencies, like for DDoS attacks that can be absorbed using multiple sites [33]. Operators may also want to control load during regular operation, perhaps because different sites have cost structures that are traffic-sensitive.

We used RIPE Atlas and Verfploeter to investigate the use of AS Prepending to adjust the catchment of a test prefix on B's sites. AS Prepending is a traffic engineering approach where an operator increases the BGP path length at one site to make that route less desirable than other routes with shorter AS paths [37]. Figure 5 shows how the distribution changes as AS prepending is applied between the two sites, as measured with both methods. (Note that the units for each measurement is different: RIPE Atlas is measured in VPs, and Verfploeter is measured in /24 blocks.) By default, with no prepending, 74% of Atlas VPs arrive at LAX, while Verfploeter shows that 78% of responsive /24 prefixes will arrive at LAX.

These results show that both measurement systems are useful to evaluate routing options. With only two sites, either measurement method seems sufficient for rough analysis. We expect the greater precision of Verfploeter will be important with more sites, and to assist with the trial-and-error process required when deploying more subtle methods of route control (for example, use of BGP communities traffic [37]).

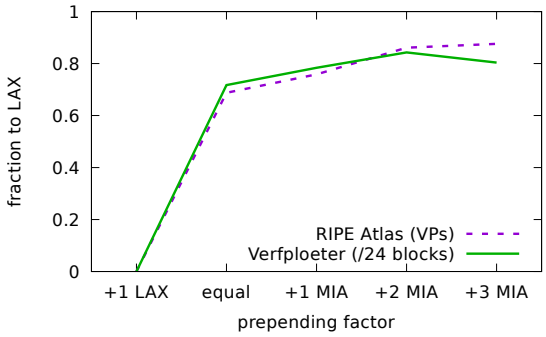


Figure 5: Split between MIA and LAX in VPs for Atlas and /24s for Verfploeter. (Dataset: *SBA-4-20, SBA-4-21, SBV-4-21.*)

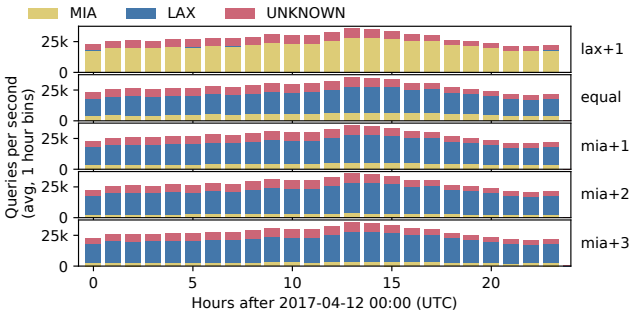


Figure 6: Predicted load for B-Root with multiple AS prepending combinations; catchment data from Verfploeter with load (Datasets: *SBV-4-21, LB-4-12*).

We next study how load shifts at different prepending values over the course of a day. For this study we measure load over 24 hours, summarizing it per hour, then combine that with measured values from five different prepending configurations (each taken once on a different day). Figure 6 shows this combination using catchment data from Verfploeter combined with DITL data of B-Root (2017-04-12). In the top graph, nearly all traffic goes to the MIA site, since LAX's BGP announcement includes an "AS prepending" of one (and the small share of load, "UNKNOWN", that is not mappable by Verfploeter). When LAX and MIA announce routes without prepending, most of the traffic load shifts to LAX (second graph from top-down). The last three graphs show the results of prepending MIA's BGP announcement by up to 3 times, resulting in an increasing traffic share shifting to LAX. However, even by announcing our prefix with 3 times our AS at MIA (MIA+3), we still see a small fraction of traffic being mapped to MIA. These few networks are likely either customers of MIA's ISP, or perhaps ASes that choose to ignore prepending.

6.2 Discovering Divisions Within ASes

Prior work (particularly anycast studies using RIPE Atlas) often assumed that anycast catchments align with ASes, thus one VP can represent where the load of the entire AS goes. While generally true for smaller ASes, this assumption is less likely to hold for large, multi-national ASes where different parts of the AS may be served by different anycast sites. Such large ASes are likely to have geographically distributed peering locations and so may prefer to direct some of their users to different anycast sites to reduce service latency.

This high density of VPs in Verfploeter allows us to test this assumption by looking for differences in anycast catchments that occur *within* individual ASes. We first remove those VPs from the dataset that show instability (see §6.3), to prevent unstable routing from being classified as a division within the AS. Without removing these VPs we observe approximately 2% more divisions (*e.g.*, ASes which are served by more than one site). We count the number of sites that are seen (from different VPs) within a single AS, in a single measurement round.

In total, we see multiple sites from 7,188 ASes, or approximately 12.7% of all ASes that were announcing at least a single prefix at the time of the measurement. Note that this is a lower-bound, using a larger and/or more diverse anycast service we might be able to determine a higher, and more accurate, percentage of ASes that are split into multiple individually routed parts.

Routing policies (like hot-potato routing) are a likely cause for these divisions. And, as routing on the Internet is largely determined by BGP, we show the number of prefixes that are announced via BGP by an AS versus the number of sites that it sees in Figure 7. Indeed, those ASes that announce more prefixes tend to see a higher amount of sites from their network.

In Figure 8 we show the number of sites that are seen from announced prefixes, grouped by prefix length. VPs in prefixes longer than a /15 are mapped to more than a single site in most cases. Even though 80% of these routed prefixes are covered by one VP (the bottom graphs in Figure 8), these are all small prefixes. About 20% of these routed prefixes are seeing more than one site and require multiple prefixes, but larger prefixes are often divided further—75% of prefixes larger than /10s see multiple sites and require multiple VPs. Although only 20% of prefixes, multiple VPs are required in prefixes that account for approximately 38% of the measured address space.

These results show that, in order to get a complete view of the catchment, in many cases you need *more* than a single VP per AS. While the quantitative results are specific to B-Root and Tangled, this qualitative result (ASes can be subdivided) applies more generally. Measurements from platforms with fewer VPs often assume that each VP can represent its AS, but likely lose precision in large ASes.

6.3 Stability of Anycast for Clients

A long-term concern with anycast is how *stable* the association of an anycast client is with its site [48]. Since TCP connections require shared state at both ends, if users switch anycast sites within the lifetime of a TCP connection, that connection will break and need to be restarted. The existence of multiple successful CDNs that use

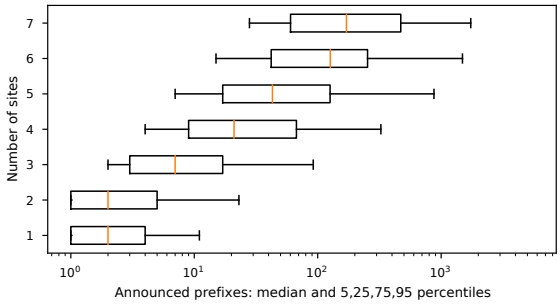


Figure 7: The number of sites that are seen from an AS versus the median amount of prefixes that are announced by those ASes. (Dataset: *STV-3-23.*)

IP anycast (including Bing, Edgecast, and Cloudflare) suggest that anycast is almost always stable, but recent work has suggested that anycast may be persistently unstable for a tiny fraction of (user, service) combinations (less than 1%) [48]. From the viewpoint of a service operator, it is interesting to know if a single measurement can be representative for a longer time, or if the catchment is continuously in flux.

Verfploeter allows us to revisit this question from Tangled to many VPs. We measured the global catchment of our testbed every 15 minutes for a day (96 observations). Considering the short-lived nature of many TCP connections this interval might be too long to detect rapid fluctuations, however, it is enough to give an impression of the overall stability of catchments. We categorize the responses (or non-responses) into 4 groups: *stable*, VPs that maintain the same catchment across measurements; *flipped*, VPs that change catchment, with responses sent to a different anycast site than the prior measurement; *to-NR*, VPs that switched to "not responding" in the current measurement; and *from-NR*, VPs that started responding in the current measurement. We do not count VPs that remain non-responsive after being counted as *to-NR*.

Figure 9 shows the results of one day of these measurement. Because the fractions of stable and flipping are so different, we break the graph into three sections. We see that the *catchment is very stable* across the measurement rounds, with a median of 3.54M (about 95% of the 3.71M that respond) VPs always replying and maintaining their prior catchment. The fraction of VPs that fluctuate between responsive and non-responsive states is small across all 96 measurements. A median of 89k (about 2.4%) VPs changed from responsive to non-responsive between measurements, and about the same number flipping back. Note that fluctuating and flipping VPs are not necessarily always the same ones.

Across the measurement period, we also see a median of 4.6k (about 0.1%) VPs change catchment (the blue line in Figure 9). All these VPs are located within 2809 ASes. Table 7 shows that 63% of the flipping VPs are part of only 5 ASes; and 51% are within AS 4134 (Chinanet). Catchment flips can be caused by changes in routing policies or link state, and frequent flipping can be caused by load balanced links. With flipping prominent in only a few ASes, these observations confirm our prior observations taken with RIPE Atlas [48], but from a larger set of vantage points: that anycast

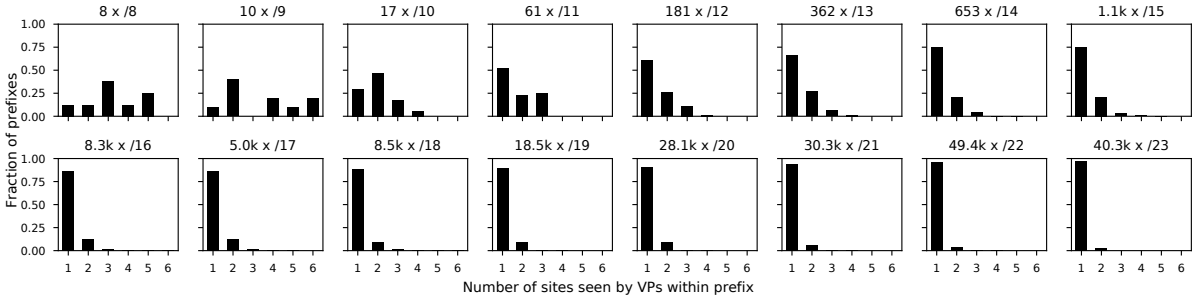


Figure 8: The number of sites that are seen for each prefix as announced in BGP. (Dataset: *STV-3-23.*)

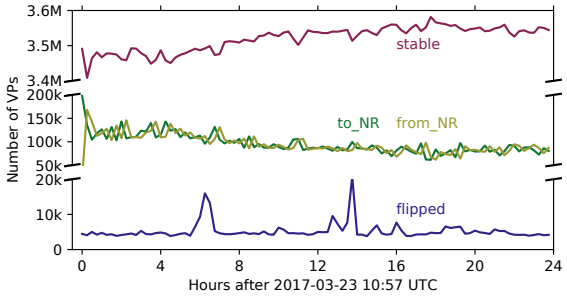


Figure 9: Stability over 24 hours. One data point per 15 minutes. If a VP stopped responding then it counts as to_NR, if it started responding it counts as from_NR. (Dataset: *STV-3-23.*)

	AS		IPs (/24s)	Flips	Frac.
1	4134	CHINANET	47,963	257,915	0.51
2	7922	COMCAST	3,933	19,133	0.04
3	6983	ITCDELTA	1,372	15,403	0.03
4	6739	ONO-AS	849	13,347	0.03
5	37963	ALIBABA	2,493	10,988	0.02
		Other	43,388	188,630	0.37
		Total	108,493	505,416	1.00

Table 7: Top ASes involved in site flips. (Dataset: *STV-3-23.*)

instability is very rare, but as a property of certain ASes, it will be persistent for users of those ASes. An additional application of Verfploeter may be identification and resolution of such instability.

7 FUTURE WORK

Although we have studied one operational service and a testbed, we are very interested in studying additional and larger services. We are also interested in looking at anycast catchments and load prediction over time, and are currently collecting this data. One could also consider approaches to improve response rate; and see if better response changes coverage estimates. These additional studies would help generalize our results, but are beyond the scope of the current paper.

We are also interested in studying CDN-based anycast systems. While the mechanics of anycast are identical regardless of the service being provided, operators of different services may optimize routing and peering differently.

Finally, it is possible that RTTs of Verfploeter measurements can be used to suggest where new anycast sites would be helpful [43] (a suggestion from an anonymous reviewer).

8 CONCLUSIONS

The key result of this paper is to show that Verfploeter allows measurements of anycast catchments across millions of networks in the Internet. Verfploeter allows us to see 430× more network blocks than RIPE Atlas, a widely used, large-scale platform for active measurements.

Such measurements are important for operating anycast services (§5.1), and more important as anycast services grow in number of sites (§5.2). With large DNS and CDN anycast networks using hundreds or thousands of sites, catchment mapping with broad coverage (§5.3) is increasingly important, particularly since regular catchment evaluation is necessary to avoid performance errors [9, 43].

Furthermore, the combination of historic traffic load and catchment mapping (§5.4) can provide a predictive tool for anycast operation (§5.5). The broad coverage of Verfploeter allows us to identify individual networks that are very likely to be the source of larger amounts of traffic.

We have used Verfploeter to understand the new B-Root anycast system (§6.1), evaluate split catchments in large ASes (§6.2), and confirm prior results in anycast stability with a larger dataset (§6.3).

ACKNOWLEDGMENTS

We thank Moritz Müller (SIDN), Giovane Moura (SIDN) and Anna Sperotto (U. Twente) for their valuable input and feedback.

Work at U. Twente was supported by SURFnet Research on Networks, SAND (http://www.sand-project.nl) and NWO DAS (http://www.das-project.nl).

Wes Hardaker's work in this paper is partially supported by USC as part of B-Root research activity.

John Heidemann's work in this paper is partially sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate (agreement FA8750-12-2-0344) and via contract number HHSP233201600010C. 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 NSF, DHS or the U.S. Government.

REFERENCES

[1] Aben, E. DNS Root Server Transparency: K-Root, Anycast and More. Web, Jan. 2017. RIPE NCC.

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

[3] Anonymous. The collateral damage of Internet censorship by DNS injection. *ACM Computer Communication Review 42*, 3 (July 2012), 21–27.

[4] Austein, R. DNS Name Server Identifier (NSID) Option. RFC 5001, August 2007.

[5] B-Root. B-Root Begins Anycast in May. http://root-servers.org/news/b-root-begins-anycast-in-may.txt, Apr. 2017.

[6] B-Root. DITL 2017 for B-Root. dataset DITL_B_Root-20170407 at https://ant.isi.edu/traces/, 2017. also available through https://www.dns-oarc.org/.

[7] B-Root. Load for 2017-05-15 for B-Root. dataset Load_B_Root-20170515 at https://ant.isi.edu/traces/, 2017.

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

[9] Bellis, R. Researching F-root anycast placement using RIPE Atlas. RIPE Labs: https://labs.ripe.net/Members/ray_bellis/researching-f-root-anycast-placement-using-ripe-atlas, Oct. 2015.

[10] 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* (Barcelona, Spain, Oct. 2013), ACM, pp. 313–326.

[11] 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* (Tokyo, Japan, Oct. 2015), ACM, pp. 531–537.

[12] Cicalese, D., Augé, J., Joumblatt, D., Friedman, T., and Rossi, D. Characterizing IPv4 Anycast Adoption and Deployment. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)* (2015), pp. 16:1–16:13.

[13] Cicalese, D., Joumblatt, D., Rossi, D., Buob, M.-O., Auge, J., and Friedman, T. A Fistful of Pings: Accurate and Lightweight Anycast Enumeration and Geolocation. In *IEEE Conference on Computer Communications (INFOCOM)* (April 2015), pp. 2776–2784.

[14] Colitti, L. K-root anycast deployment. Presentaiton at RIPE-52, April 2006.

[15] Danzig, P. B., Obraczka, K., and Kumar, A. An analysis of wide-area name server traffic: A study of the Domain Name System. In *Proceedings of the ACM SIGCOMM Conference* (Jan. 1992), pp. 281–292.

[16] DNS-OARC. Day In The Life of the internet (DITL) 2017. https://www.dns-oarc.net/oarc/data/ditl/2017, Apr. 2017.

[17] Fan, X., and Heidemann, J. Selecting representative IP addresses for Internet topology studies. In *Proceedings of the ACM Internet Measurement Conference* (Melbourne, Australia, Nov. 2010), ACM, pp. 411–423.

[18] Fan, X., Heidemann, J., and Govindan, R. Evaluating anycast in the Domain Name System. In *Proceedings of the IEEE Infocom* (Turin, Italy, Apr. 2013), IEEE, pp. 1681–1689.

[19] Flavel, A., Mani, P., Maltz, D. A., Holt, N., Liu, J., Chen, Y., and Surmachev, O. FastRoute: A Scalable Load-Aware Anycast Routing Architecture for Modern CDNs. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI)* (2015), pp. 381–394.

[20] Fomenkov, M., k. c. claffy, Huffaker, B., and Moore, D. Macroscopic internet topology and performance measurements from the DNS root name servers. In *Proceedings of the USENIX Large Installation Systems Administration Conference* (San Diego, CA, USA, Dec. 2001), USENIX, pp. 221–230.

[21] Gill, P., Crete-Nishihata, M., Dalek, J., Goldberg, S., Senft, A., and Wiseman, G. Characterizing web censorship worldwide: Another look at the opennet initiative data. *ACM Transactions on the Web 9*, 1 (Jan. 2015).

[22] Giordano, D., Cicalese, D., Finamore, A., Mellia, M., Munafã, M., Joumblatt, D. Z., and Rossi, D. A first characterization of anycast traffic from passive traces. In *IFIP workshop on Traffic Monitoring and Analysis (TMA)* (April 2016), pp. 30–38.

[23] Grubb, B. 8.8.8.8: The four digits that could thwart Australia's anti-piracy, website-blocking regime. *Sydney Morning Herald* (June 24 2015).

[24] Hardaker, W. Verfploeter measurements of B-Root. USC-LANDER dataset B-Root_Verfploeter-20170421 and B-Root_Verfploeter-20170515, https://ant.isi.edu/datasets/, 2017.

[25] Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., and Bannister, J. Census and survey of the visible Internet. In *Proceedings of the ACM Internet Measurement Conference* (Vouliagmeni, Greece, Oct. 2008), ACM, pp. 169–182.

[26] Hussain, A., Bartlett, G., Pryadkin, Y., Heidemann, J., Papadopoulos, C., and Bannister, J. Experiences with a continuous network tracing infrastructure. Tech. Rep. ISI-TR-2005-601, USC/Information Sciences Institute, Apr. 2005.

[27] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. Internet inter-domain traffic. In *Proceedings of the ACM SIGCOMM Conference* (New Delhi, India, Aug. 2010), ACM, pp. 75–86.

[28] Liang, J., Li, J. J. H. D. K., and Wu, J. Measuring query latency of top level DNS servers. In *Proceedings of the Passive and Active Measurement Workshop* (Hong Kong, China, Mar. 2013), Springer, pp. 145–154.

[29] Madory, D. Iran Leaks Censorship via BGP Hijacks. Web, Jan. 2017. Dyn.

[30] Madory, D., Popescu, A., and Zmijewski, E. Accidentally Importing Censorship - The I-root instance in China. Presentation, June 2010. Renesys Corporation.

[31] Mauch, J. Open resolver project. Talk at DNS-OARC, May 2013.

[32] MaxMind. MaxMind GeoLite2. http://dev.maxmind.com/geoip/geoip2/geolite2/.

[33] Moura, G. C. M., de O. Schmidt, R., Heidemann, J., de Vries, W. B., Müller, M., Wei, L., and Hesselman, C. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In *Proceedings of the ACM Internet Measurement Conference* (Nov. 2016), pp. 255–270.

[34] PlanetLab. https://www.planet-lab.org/.

[35] Poese, I., Uhlig, S., Kaafar, M. A., Donnet, B., and Gueye, B. IP geolocation databases: Unreliable? *SIGCOMM Comput. Commun. Rev. 41*, 2 (Apr. 2011), 53–56.

[36] Quan, L., Heidemann, J., and Pradkin, Y. Trinocular: Understanding Internet reliability through adaptive probing. In *Proceedings of the ACM SIGCOMM Conference* (Hong Kong, China, Aug. 2013), ACM, pp. 255–266.

[37] Quoitin, B., Pelsser, C., Bonaventure, O., and Uhlig, S. A performance evaluation of BGP-based traffic engineering. *International Journal of Nework Management 15*, 3 (May 2005), 177–191.

[38] RIPE. RIPE Atlas measurements of B-Root. https://atlas.ripe.net/measurements/, measurement IDs 10310, 8310594, 8312460, 8312974, 8313009, 8313262, 2017.

[39] RIPE NCC. DNSMON. https://atlas.ripe.net/dnsmon/.

[40] RIPE NCC Staff. RIPE Atlas: A Global Internet Measurement Network. *The Internet Protocol Journal 18*, 3 (Sept. 2015), 2–26.

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

[42] RSSAC. RSSAC advisory on measurements of the root server system. Tech. Rep. RSSAC002v3, ICANN, June 2016.

[43] Schmidt, R. d. O., Heidemann, J., and Kuipers, J. H. Anycast latency: How many sites are enough? In *Proceedings of the Passive and Active Measurement Workshop* (Sydney, Australia, Mar. 2017), Springer, pp. 188–200.

[44] Thousand Eyes. DNS monitoring service. Service description at https://www.thousandeyes.com/solutions/dns, 2017.

[45] Vixie, P., and Kato, A. DNS referral response size issues. Work in progress (Internet draft draft-ietf-dnsop-respsize-14, May 2012.

[46] de Vries, W. RIPE Atlas measurements of Tangled. Tangled_Atlas-20170201 https://traces.simpleweb.org/ and RIPE Atlas measurement IDs 7794356-7794364, 2017.

[47] de Vries, W. Verfploeter measurements of Tangled. dataset Tangled_Verfploeter-20170201 and Tangled_Verfploeter-20170323, https://traces.simpleweb.org/, 2017.

[48] Wei, L., and Heidemann, J. Does anycast hang up on you? In *IEEE International Workshop on Traffic Monitoring and Analysis* (Dublin, Ireland, July 2017), IEEE, p. to appear.

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

[50] Yurdam, S. Using graffiti, Turks share tips for getting around Twitter ban. *The Observers* (21 March 2014).