Old but Gold: Prospecting TCP to Engineer DNS Anycast (extended)

USC/ISI Technical Report ISI-TR-740

June 2020

Giovane C. M. Moura (1) John Heidemann (2) Wes Hardaker (2)
Jeroen Bulten (3) Joao Ceron (1) Cristian Hesselman (1,4)
1: SIDN Labs 2: USC/ISI 3: SIDN 4: University of Twente

ABSTRACT

DNS latency is a concern for many service operators: CDNs exist to reduce service latency to end-users, but must rely on global DNS for reachability and load-balancing. We show that a recursive DNS resolver's preference for low latency shifts traffic at TLDs and the DNS root. DNS latency today is monitored with distributed infrastructure such as RIPE Atlas, or with active probing using Verfploeter. While Atlas coverage is wide, it is incomplete, and Verfploeter coverage in IPv6 is limited. In this paper we show that *passive observation of TCP handshakes provides a mechanism to measure DNS latency*. Passive RTT estimation from TCP is an old idea, but it has never been used to examine DNS before. We show that there is sufficient TCP DNS traffic today to provide greater coverage than existing approaches, and is the best method to observe latency of DNS using IPv6. We show that estimates of DNS latency from TCP is consistent with UDP latency. Our approach finds real problems: We define *DNS polarization*, a new problem where a hypergiant sends global traffic to one anycast site rather than taking advantage of the global anycast deployment—we found Google traffic polarized and cut its latency from 100 ms to 10 ms, and for Microsoft, the latency cut due to traffic being depolarized was from 90 ms to 20 ms. Our approach is in operational use for a European country's top-level domain, and monitoring with our tool helped find and correct a routing detour sending European traffic to Australia.

KEYWORDS

DNS, recursive DNS servers, TCP, RTT, Anycast, BGP, Routing

1 INTRODUCTION

Latency is a key performance indicator for many DNS operators. DNS latency is seen as a bottleneck in web access [45]. Content Delivery Networks (CDNs) are particularly sensitive to DNS latency because, although DNS uses caching extensively to avoid latency, many CDNs use very short DNS cache lifetimes to give frequent opportunities for DNS-based load balancing and replica selection [12]. As a result of operator attention to DNS latency, low latency is a selling point for many commercial DNS operators, many of whom deploy extensive distributed systems with tens, hundreds, or more than 1000 sites [8].

DNS deployments often use *IP anycast* [21, 30] to reduce latency for clients. A *DNS service* is typically provided by two or more *authoritative DNS servers* [15], each defined as DNS on a separate IP address (in the NS record set [23]). With IP anycast, the IP address assigned to the authoritative DNS server is announced from many physically distributed *sites*, and BGP selects which clients go to which site.

DNS clients often select the lowest-latency authoritative server when they have a choice [25, 26]. We will show later (§5) that improving anycast latency shifts traffic loads between servers.

DNS latency has been extensively studied [9, 24, 42]. Previous studies have looked at both absolute latency [42] and how closely it approaches speed-of-light optimal [19, 44]. A number of papers measure DNS latency from measurement systems with distributed vantage points such as RIPE Atlas [36], sometimes to optimize latency [7, 22]. Recent work has shown how to measure anycast catchments with active probes with Verfploeter [10, 11], one application of which is measuring latency. However, both of these approaches to measure latency provide mixed coverage: large hardware-based measurements like RIPE Atlas only have about 11k active vantage points and cover only 8670 /24 IPv4 network prefixes [34] (May 2020). Verfploeter provides much better coverage, reaching millions of networks, but it depends on a response from its targets and so cannot cover networks with commonly deployed ICMP-blocking firewalls. It is also difficult to apply to IPv6 since it requires a target list, and effective IPv6 hitlists are an open research problem

The main contribution of this paper to show how passive analysis of the TCP connection setup's *handshake latency* can measure DNS client latency. The TCP handshake has been used to estimate RTT at endpoints since 1996 [14], and it is widely used in passive analysis of HTTP (for example, [40]). This paper is the first to describe the technique's application

to DNS. *Passive RTT estimation can increase coverage* beyond current techniques that measure DNS latency: it provides coverage of exactly the customers that are using the DNS service being analyzed, and it is the only current approach that can provide coverage for IPv6 networks when many hosts use private stateless IPv6 addresses [29].

Our second contribution is to show that *TCP-handshakes provide effective estimation* of DNS latency. Although DNS most often uses UDP, leaving DNS-over-TCP (shortened to DNS/TCP) to be often *overlooked*, we show that there is enough DNS/TCP traffic to support good coverage of latency estimation (§2.1). We show that if we prospect through it we can find the latency "gold". We also show measured latency from UDP and our estimates from TCP are similar. We have added TCP analysis to *ENTRADA* [43, 48], an open source DNS analysis platform.

Our final contribution is to show that *TCP-based latency estimation matters*—it detects latency problems in operational networks (§4). Working with an European Country-code top-level domain (`.nl` ccTLD) and two commercial DNS providers, we found two cases of *DNS polarization*, an interaction between Google and Microsoft and anycast operators. These companies are Internet "hypergiants"[31] with their own backbones, yet we found they each sent global traffic to just one location, resulting in latency inflation of 150 ms for many clients. Second, we show that passive analysis of DNS/TCP is lightweight enough to run 24x7 for problem detection. In one event, we found large increases in RTT for some networks, a problem we traced to mis-routing that sent traffic from Europe to anycast in Australia (§4.4).

2 DNS/TCP FOR RTTS?

While UDP is the preferred transport layer for DNS, TCP support has always been required to handle large replies [6]. TCP has also always been used for zone transfers between servers, and now increasing numbers of clients are using TCP in response to DNSSEC [1], response-rate limiting [46], and recently DNS privacy [16].

The RTT between a TCP client and server can be measured passively during the TCP session establishment [14, 22] or during the connection teardown [40]. For passive TCP observations to support evaluation of anycast networks for DNS, (a) enough clients must send DNS over TCP so they can serve as *vantage points* (VPs) to measure RTT, and (b) the RTT for queries sent over TCP and UDP should be the same.

We next verify these two requirements, determining how many clients can serve as VPs with data from three production authoritative severs (§2.1) – two from the `.nl` zone, and `B-root`, one of the Root DNS servers [39]. We then compare the RTT of more than 10k VPs with both TCP and UDP to confirm they are similar (§2.2).

2.1 Does DNS/TCP provide Enough Coverage?

To assess whether DNS/TCP has enough coverage in production authoritative servers, we look at production traffic of two DNS zones: `.nl` and the DNS Root. For each zone we measure: (a) the number of resolvers using the service; (b) the number of ASes sending traffic; (c) the fraction of TCP queries the servers receive; (d) the percentage of resolvers using both UDP and TCP; and (e) the RTT of the TCP packets.

Our goal is to get a good estimate of RTT latency that covers every client's network. If every query were TCP, we could determine the latency of each query and get 100% coverage. However, most DNS queries are sent over UDP instead of TCP. We, therefore, look for *representation*—if we have a measured query over TCP, is its RTT the same as the RTTs other queries that use UDP, or that are from other nearby resolvers? If network conditions are relatively stable, the TCP query's RTT can represent the RTT for earlier or later UDP queries from the same resolver. It will likely represent the RTT for other resolvers in the same /24 as well. It is even possible it may represent the RTT for other resolvers in the same AS. Each assumption gives us greater coverage but increases the chances that the TCP RTT differs from the RTT of the other queries.

2.1.1 `.nl` *authoritative servers.* `.nl` currently (Oct. 2019) has 4 Authoritative DNS services, all configured with IP anycast. We next examine data from two of four authoritative services, called here Anycast Services A and B. The anycast services consists of 6 and 18 sites distributed globally. Each is run by a third-party DNS operator, one headquartered in Europe and the other in North America. They have no joint commercial relationship and we believe they have disjoint service infrastructure.

We analyze one week of traffic (2019-10-15 to -22) for each services using *ENTRADA*. That week from each service handles about 10.9 billion queries from about 200k resolvers and 50k Autonomous Systems (ASes), as can be seen in Table 1.

This week of data shows that TCP used rarely, less than 7% of queries each anycast service. However, those queries can represent more than a fifth of resolvers and 44% of ASes.

It may see that 44% of ASes is insufficient coverage to understand anycast performance. We believe this coverage is meaningful because it includes data for the ASes that send the most data, and query distribution per ASes is heavily skewed towards a few "heavy hitters". Figure 1 shows that the top 10 ASes are responsible for about half of all queries, while the top 100 are responsible 78% and 75% of all queries, for Services A and B.

It is often appropriate for one TCP to represent its AS. For ASes where all recursive resolvers are co-located, latency to one is the same as to the others, so this assumption

	Queries		Resolvers		ASes	
	Anycast A	**Anycast B**	**Anycast A**	**Anycast B**	**Anycast A**	**Anycast B**
Total	5 237 454 456	5 679 361 857	2 015 915	2 005 855	42 253	42 181
IPv4	4 005 046 701	4 245 504 907	1 815 519	1 806 863	41 957	41 891
UDP	3 813 642 861	4 128 517 823	1 812 741	1 804 405	41 947	41 882
TCP	191 403 840	116 987 084	392 434	364 050	18 784	18 252
ratio TCP	5.02%	2.83%	21.65%	20.18%	44.78%	43.58%
IPv6	1 232 407 755	1 433 856 950	200 396	198 992	7 664	7 479
UDP	1 160 414 491	1 397 068 097	200 069	198 701	7 662	7 478
TCP	71 993 264	36 788 853	47 627	4 6190	3 391	3 354
ratio TCP	6.2%	2.63%	23.81%	23.25%	44.26%	44.85%

Table 1: DNS usage for two authoritative services of `.nl` (Oct. 15–22, 2019).

	Anycast A	**Anycast B**
IPv4	4 005 046 701	4 245 504 907
from TCP ASes	3 926 025 752	4 036 328 314
Ratio (%)	98.02%	95.07%
from TCP resolvers	2 306 027 922	1 246 213 577
Ratio (%)	57.7%	29.35%
IPv6	1 232 407 755	1 433 856 950
from TCP ASes	1 210 649 060	1 386 035 175
Ratio (%)	98.23%	96.66%
from TCP resolvers	533 519 527	518 144 495
Ratio (%)	43.29%	36.13%

Table 2: Queries per Services for ASes and Resolvers that send TCP queries for `.nl` (Oct. 15–22, 2019).

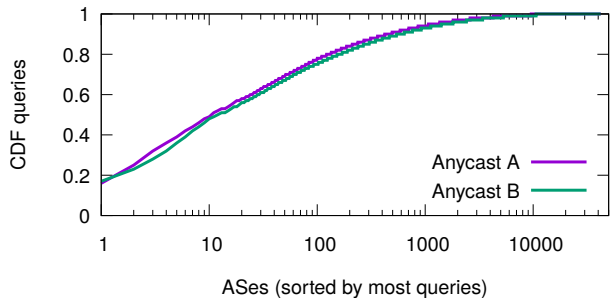


Figure 1: `.nl`: queries distribution per AS.

is appropriate. Table 2 shows coverage using this assumption: the relatively few number of TCP queries can represent 95–98% of all traffic under this assumption. Furthermore, because most queries come from a few ASes and their resolvers, we have TCP traffic from resolvers covering 29-57% of all queries.

With regards to /24 IPv4 prefixes, we notice that 770k and 769 k send queries to Anycast A and B, respectively. Out of these, 141k and 132k send queries over TCP.

Root DNS. To confirm that DNS/TCP provides coverage beyond `.nl`, we also look at how many TCP queries are seen at most Root DNS servers [39] over the same period.

Table 3 shows RSSAC-002 statistics [17, 47] from 11 of the 13 Root DNS services reporting at this time. As can be seen, the ratio of TCP traffic varied for each service (known as "letters", from A to M) and IPv4 or IPv6, overall ranging from 2.8 (A Root over IPv4) up 18.9% (J Root over IPv6). This data suggests the root letters seem similar DNS/TCP rates as `.nl`.

Inducing coverage. While TCP coverage is not complete, we can get complete coverage by actively managing traffic to induce occasional TCP queries, as is often done in web systems (for example [41]). The DNS specification includes the TC bit to indicate a truncated reply that must be retried over TCP. DNS Receiver Rate Limiting [46] uses this mechanism to force possible UDP-based address spoofers to resend their queries with TCP, allowing TCP cookies to source-address prevent spoofing. We can use the same mechanism to a small number (perhaps 0.1%) of all queries to force them to switch from UDP to TCP, allowing us to determine RTTs.

Summary: We see that TCP data can represent a majority of all queries made to this service. More importantly, TCP provides the *only* insight into IPv6 latency, since current active methods do not generalize to IPv6.

2.2 DNS/UDP vs. DNS/TCP RTT

We expect round-trip-times to be the same measured with DNS/TCP and DNS/UDP. We next confirm that assumption.

We can compare DNS/UDP and DNS/TCP RTTs by comparing *query response times* and accounting for TCP connection setup. DNS/UDP makes a direct request and gets a response, while in DNS/TCP we set up the TCP connection (with a SYN–SYN/ACK handshake), so a TCP DNS request should take two RTTs (assuming no connection reuse, TCP fast-open, or other optimizations). After accounting for TCP's handshake we should get similar RTT estimates.

	A	B	C	D	F	H	I	J	K	L	M
Total	70601	40601	59033	88136	144635	31702	66582	115162	76761	105041	42702
IPv4	58552	33925	47675	74565	125020	25706	55874	96727	61378	88046	33687
UDP	56921	32334	45568	70969	118738	25234	51208	87891	60312	84059	31925
TCP	1631	1591	2107	3596	6282	472	4665	8836	1065	3986	1762
Ratio (TCP)	2.87%	4.92%	4.62%	5.07%	5.29%	1.87%	9.11%	10.05%	1.77%	4.74%	5.52%
IPv6	12049	6675	11357	13571	19614	5995	1070	18435	15383	16994	9014
UDP	11659	6280	10966	13071	18919	5825	936	15511	15108	16576	8268
TCP	389	394	391	499	694	169	1342	2923	274	418	746
Ratio TCP	3.34%	6.29%	3.57%	3.82%	3.67%	2.92%	14.34%	18.84%	1.82%	2.52%	9.03%

Table 3: DNS queries (in millions) for Root DNS (E and G missing) – 2019-10-15 – 2019-10-22.

	DNS/UDP	DNS/TCP	Ratio
Matching VPs	10129		
Queries	102537	109161	
RTT			
median	30.06	60.26	2.00
75%ile	31.42	63.38	2.00
90%ile	33.13	66.79	2.00

Table 4: DNS/UDP and DNS/TCP Query Response times distribution for ns3.dns.nl [33].

To confirm this claim we measure DNS/UDP and DNS/TCP query response times using RIPE Atlas [35]. Atlas provides about 11k devices in different locations around the world, allowing us to test many network conditions. We send queries to one authoritative server: `ns3.dns.nl`, one of the production authoritative servers for `.nl`. We send queries with both protocols from 10,129 Vantage Points (VPs), each VP being an Atlas probe and its local resolver active in both measurements. We configure the measurements to run for 1 hour, sending queries every 10 minutes. In total, we observe more than 200k queries.

Table 4 shows the response times for all measurements. We see that DNS/TCP consistently takes twice as long as UDP, as predicted, even at different points of the distribution. This experiment proves that passively observed TCP RTTs can represent the RTTs that DNS/UDP will see.

3 PRIORITIZING ANALYSIS

We have shown that DNS/TCP can be mined to determine RTTs (§2). Operational DNS systems must serve the whole world, there are more than 30k active ASes sending DNS queries to authoritative servers. Both detection and resolution of networking problems in anycast systems is labor intensive: detection requires both identifying specific problems and their potential root causes. Problem resolution requires new site deployments or routing changes, both needing human-in-the-loop changes involving trouble tickets, new hardware, and new hosting contracts.

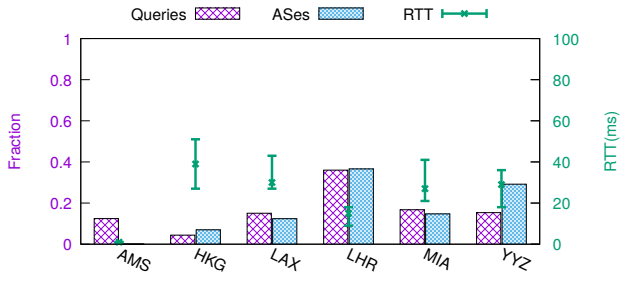
Overview: We use two strategies to *prioritize* analysis of problems that are most important: per-anycast site analysis and per client AS analysis, and rank each by median latency, interquartile range (IQR) of latency, and query volume.

Studying anycast sites focus on "our" side of the problem, highlighting locations in the anycast service we are responsible for that shows high latency toward sites, drawing our attention. Fortunately, because we are responsible for the sites operating our DNS service, we often have some control over how they peer.

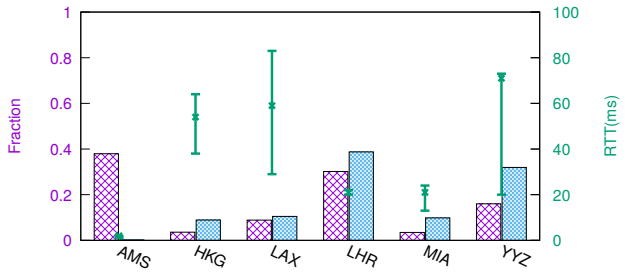
Clients ASes examine the *user* side of the problem (at recursive resolvers), since client latency is a goal in DNS service. While performance in client ASes can be difficult to improve because we do not have a direct relationship with those network operators, we show in §4 that we can address problems in some cases.

Finally, we consider median latency, interquartile range, and query volume to prioritize investigation. Median latency is a proxy for overall latency at the site. Interquartile range, the difference between 75%ile and 25%ile latencies, captures the *spread* of possible latencies at a given site or AS. Finally, query volume (or rate) identifies locations where improvements will affect more users. We sort by overall rate rather than the number of unique sources to prioritize large ASes that send many users through a few recursive resolvers (high rate, low number of recursive IPs).

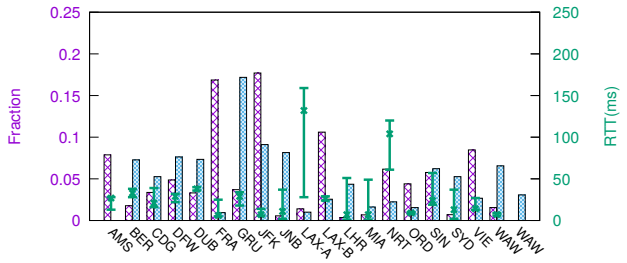
Prioritization by Site: Figure 2 shows per-site latency for `.nl`, broken out by protocol (IPv4 and IPv6) and by site, for two anycast services (A and B). For each site, we show two bars: the fraction of total queries and number of ASes (filled and hatched bar in each cluster). We overlay both with whiskers for latency (with median in the middle and



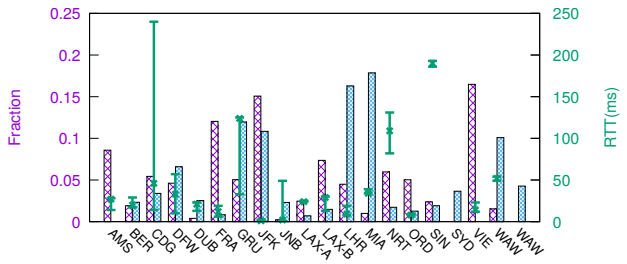
(a) Anycast A: IPv4



(b) Anycast A: IPv6



(c) Anycast B: IPv4



(d) Anycast B: IPv6

Figure 2: `.nl` distribution of queries and ASes per site (pink bars) and latency (median, 25%ile, and 75%ile, (green lines), for each anycast site, for two services (Anycast A and B) and two protocols (IPv4 and IPv6). Data from 2020-10-15 to -22.

25%ile and 75%ile at whisker ends). In these graphs some sites (such as CDG for Anycast B in IPv6) stand out with high interquartile ranges, while others with lower interquartile range (for Anycast B, LAX-A and NRT in IPv4 and NRT and GRU in IPv6).We look at these cases in detail in §4.

We omit graphs for `B-root` by site due for anonymization.

Prioritization by Client AS: Figure 3 and Figure 4 show the distribution of latency for the top-ten ASes with largest query volume for Anycast A and B of `.nl`. While many ASes show good latency (low median and small interquartile range), we see the top two busiest ASes for Anycast A in IPv4 (Figure 3a) show a high median and large interquartile range (Figure 3b). These ASes experience anycast polarization, a problem we describe in §4.3.

Figure 5 shows latencies for the top ASes for `B-root`. Here we show quartile ranges as boxes, and with the 10%ile and 90%ile values as whiskers. Rather than split by protocol, here we show both rankings (Figure 5a) and by query rate (Figure 5b) on the x-axis. While rank gives a strict priority, showing ASes by rate helps evaluate how important it is to look at more ASes (if the next AS to consider is much, much lower rate, addressing problems there will not make as large a difference to users).

Finally, we find that an AS' inter-quartile range (IQR) often highlights ASes that show routing problems. Figure 6 sorts ASes by IQR. We limit the vertical scale, since 90%iles of some ASes can be quite large (multiple seconds).

We identify specific problems from in these graphs next.

4 PROBLEMS AND SOLUTIONS

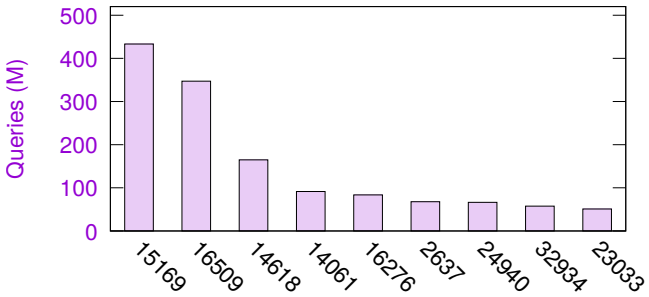
Given new information about both IPv4 and IPv6 latency from DNS/TCP (§2), and priorities (§3), we next examined anycast performance for two of the four anycast services operating for `.nl`, and for `B-root`. For each problem we describe how we found it, the root causes, and, when possible, solutions and outcomes.

4.1 Distant Lands

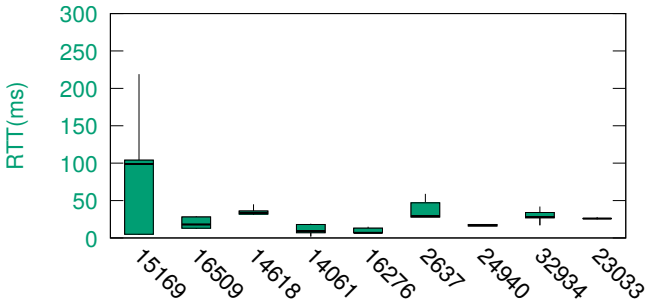
The first problem be describe is *distant lands:* when a country has no anycast server locally and has limited connectivity to the rest of the world. When trans-Pacific traffic was metered, these problems problem occurred for Australia and New Zealand. Today we see this problem with China, China has a huge population of Internet users, but its international network connections that can exhibit congestion [49].

Detection: We discovered this problem by observing large interquartile latency for `.nl`'s Anycast B in v4 (Figure 2c) and v6 (Figure 2d) at Tokyo (NRT, both v4 and v6), Singapore (SIN, v6), and CDG (v6), all with 75%iles over 100 ms.

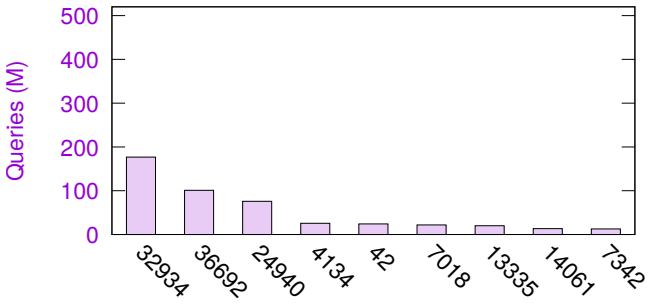
These wide ranges of latency prompted us to examine which recursive resolvers visiting these sites and showed high latency. Many queries come from ASes in Asia (Figure 7).



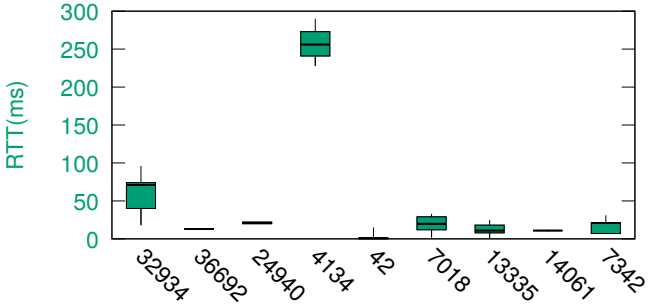
(a) Anycast A – IPv4 – Queries



(b) Anycast A – IPv4 – RTT

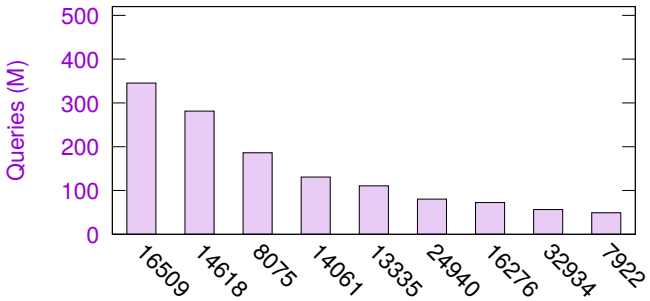


(c) Anycast A – IPv6 – Queries

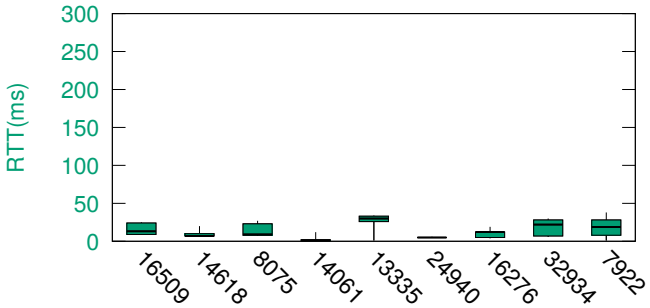


(d) Anycast A – IPv6 – RTT

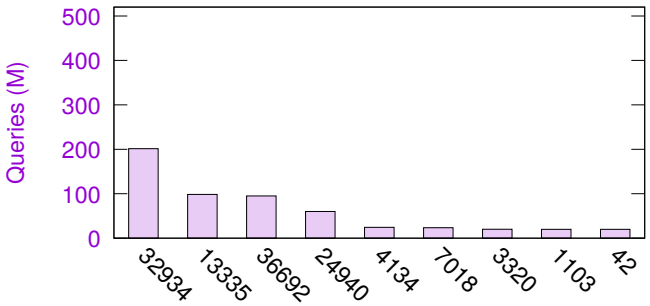
Figure 3: `.nl` Anycast A queries and RTT for the 10 ASes ranked by most queries (bars left axis). Data; 2019-10-15 to -22.



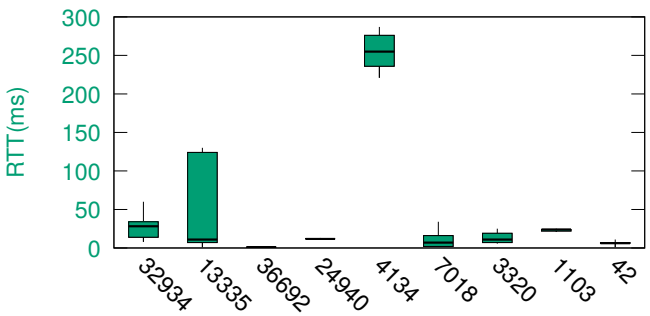
(a) Anycast B – IPv4 – Queries



(b) Anycast B – IPv4 – RTT

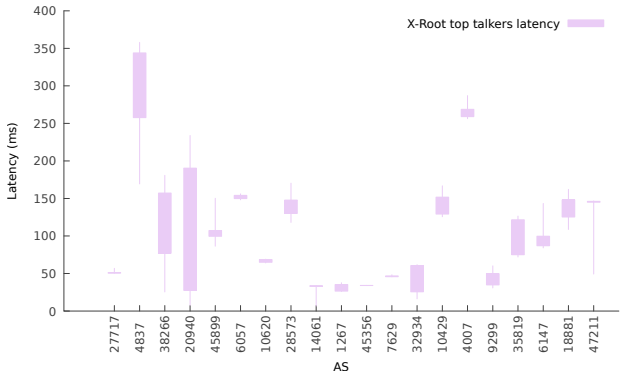


(c) Anycast B – IPv6 – Queries

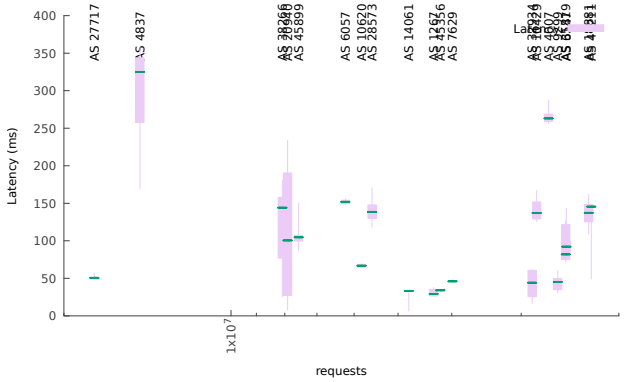


(d) Anycast B – IPv6 – RTT

Figure 4: `.nl` Anycast B query RTT for the 10 ASes ranked by most queries (bars left axis). Data: 2019-10-15 to -22.

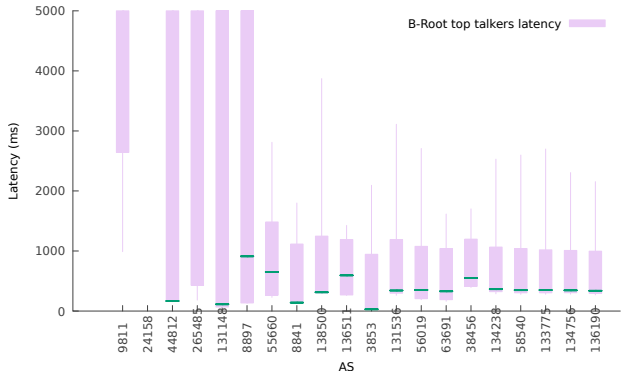


(a) `B-root` top talkers latency by rank

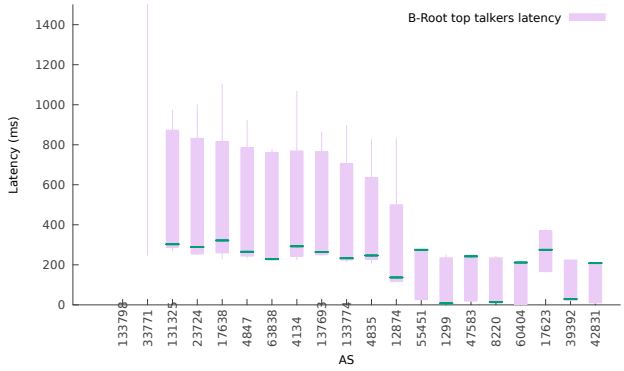


(b) `B-root` top talkers latency by data size

Figure 5: Latency analysis to `B-root` by AS traffic volume – AS list is in Table 7



(a) `B-root` AS latency diversity for IPv4 by rank



(b) `B-root` latency diversity for IPv6 by rank

Figure 6: Latency analysis to `B-root` by AS latency diversity.

NRT sees many queries (6.1% of total, more than it's "fair share" of 5.2%). Of the top 10 ASes sending queries to NRT, 9 are from China (see Figure 7).

We see a number of Chinese ISPs also send IPv6 traffic to Paris (CDG), resulting in it's wide spread of RTTs. Not only must the traverse congested international links, but they then travel to a geographically distant anycast site, raising the 75%ile RTT at CDG over 100 ms (even though its median is under 22 ms).

Resolution: While we can diagnose this problem, the best resolution would be new anycast servers for Anycast B inside China. The operator is working on deploying in China, but only recently have foreign DNS providers been allowed to operate locally there [49].

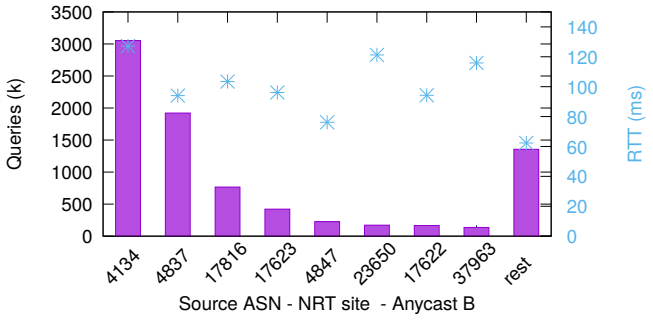


Figure 7: Anycast B, Japan site (NRT): Top 8 querying ASes are Chinese, and responsible for 80% of queries.

4.2 Prefer-Customer to Another Continent

The second root-cause problem we found is when one AS prefers a distant anycast site, often on another continent, because that site is a customer of the AS. (Recall that a common BGP routing policy is *prefer customer:* if an AS can satisfy a route through one of its customers, it prefers that choice over an alternate route through a peer or transit provider. Presumably the customer is paying the AS for service, while sending the traffic to a peer or via transit is either cost-neutral or incurs additional cost.)

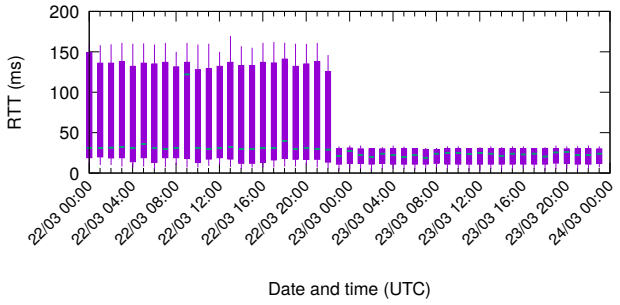


Figure 8: Anycast B and Comcast: RTT before and after resolving IPv6 misconfiguration.

	Queries	Queries Top Site	(% top site)
Google	860 775 677	860 774 158	99.9998
IPv4	433 145 168	433 145 119	99.9999
IPv6	427 630 509	427 629 039	99.9997
Microsoft	449 460 715	449 455 487	99.9988
IPv4	449 439 957	449 434 729	99.9988
IPv6	20 758	20 758	100

Table 5: Anycast A: Polarized ASes and query distribution (Oct 15-22,2019).

We have seen this problem in two situations, at `.nl` Anycast B's Brazil site, and with `B-root` for its site in South America.

`.nl` Detection: We detected this problem for `.nl` Service B by observing high IPv6 median latency (124 ms) for queries is in São Paulo, Brazil (GRU) in Figure 2d. Examination of the data shows many of the high-latency queries are from Comcast (AS7922), a large U.S.-based ISP. As with China and CDG, this case is an example of queries traveling out of the way to a distant anycast site, ignoring several anycast sites already in North America.

We confirmed that North American clients of this AS were routing to the Brazil site by checking CHAOS TXT queries [3] from RIPE Atlas probes to Anycast B (data: ComcastV6 [33]).

`.nl` Resolution: We contacted `.nl` Anycast B's operator who identified that the issue was that one of their upstream providers. This provider had deployed BGP communities to limit the IPv4 route to South America. After our contact, they deployed the same community for IPv6 and the Comcast traffic remained in the US.

We first confirm the problem was resolved by analyzing traces from Anycast B, and by confirming that Comcast IPv6 clients were now answered by other North American sites. The solution reduced 75%ile latency by 100 ms: in Figure 8 before the change, IPv6 shows IQR of 120 ms for Anycast B. After this change on 2020-03-23t00:00, we see the IQR falls to 20 ms.

Second, we also verified with Atlas probes hosted on Comcast's network (data: ComcastV6-afterReport in [33]), and the median RTT from Comcast Atlas was reduced from 139 ms to 28 ms.

This operational problem was found and resolved due to this paper's DNS/TCP analysis.

`B-root` Detection: `B-root` has observed high latencies for traffic going to a South-American anycast site of `B-root`. As with `.nl` and GRU, we examined traffic and identified a primarily-North American ISP that was sending all of its traffic to the South American site, ignoring all other lower-latency sites. We then confirmed that an AS purchases transit from this ISP.

`B-root` Resolution: We do not yet have a completely satisfactory resolution to this problem. Unfortunately the AS that purchases transit from the North American ISP does not directly peer with `B-root`, so we cannot control its peering. We currently poison the route to prevent latency problems, but that greatly reduces traffic arriving at this site.

4.3 Anycast Polarization with Google and Microsoft

We next describe *anycast polarization*, a problem we believe has not been previously described. Like prefer-customer, it involves high latency that derives from traffic being needlessly sent to another continent. But it follows from BGP's limited knowledge of latency (AS path length is its only distance metric) and the flattening of the Internet [18].

4.3.1 Detecting the Problem. We discovered this problem by examining DNS/TCP-derived latency from the two largest ASes sending queries to `.nl` Anycast A. As seen in Figure 3a and Figure 3c, AS8075 (Microsoft) and AS15169 (Google) show very high IPv4 median latency (74 ms and 99 ms), and Google shows a very high IQR (99 ms) Google also shows a high IPv6 median latency (104 ms).

Both Google and Microsoft are hypergiants [31], with datacenters on multiple continents. Both also operate their own international backbones and peer with the Internet in dozens of locations. These very high latencies suggest much of their DNS traffic is traveling between continents and not taking advantage of `.nl`'s global anycast infrastructure.

Confirming the problem: `.nl` Anycast A has six sites, so we first examine how many queries go to each site. Table 5 show the results—all or very nearly all (four or five "nines") go to the a single anycast site due to routing preferences. For Google, this site is in Amsterdam, and for Microsoft, Miami.

While a preferred site is not a problem for a small ISP in one location, it is the root cause of very high latency for these hypergiants. They are routing their global traffic over

their own backbones to one physical location. Even if it is the best destination for some of their traffic, one location can never be the lowest latency for all globally distributed datacenters. They defeat any advantages anycast has for reducing latency [26, 42].

4.3.2 Depolarizing Google to `.nl` *Anycast A. Root-cause of polarization:* We first investigated Google's preference for AMS. `.nl` directly operates the AMS site (the other 5 sites are operated by a North American DNS provider). We determined (working with both the AMS and Google operators) that Google has a direct BGP peering with the site at AMS. BGP prefers routes with the shortest AS-PATH, and in addition, ASes often prefer Private Network Interconnect (PNIs) over equal length paths through IXPs, so it is not surprising it prefers this path. (The general problem of BGP policy interfering with lowest latency is well documented [4, 5, 7, 19, 22, 42]. We believe we are the first to document this problem with hypergiants and anycast through PNI.)

We next describe how we worked with the AMS operators and Google to resolve this problem. We document this case as one typical resolution to show the need for continuous observation of DNS latency through DNS/TCP not find the problem and confirm the fix.

Figure 9 show the effects of our traffic engineering on anycast use and query latency for both IPv4 and IPv6. Each graph shows traffic or median client latency for each of the 6 `.nl` Anycast A sites. (Query latency is determined by DNS/TCP traffic over each day.) The graphs show behavior over January 2020, January 5th to 9th (the left, pink area) before any changes, the 9th to the 21st (the middle, green area) when the AMS route was withdrawn, and finally after the 21st (the right, blue region) when AMS was restored, but with different kinds of policy routing.

These graphs confirm that AMS received all traffic from Google initially, causing Anycast A to be experienced by Google as an unicast service. We see that the median latency for Google about 100 ms, a large value made worse given Google sends most queries to this service (Figure 3a). Withdrawing the AMS peering with Google corrects the problem with queries now sent to most sites, and, as such, enabling Google to take advantage of the anycast configuration of the service. We see median latency dropping to 10 to 40 ms, although still around 100 ms at YYZ in Toronto, Canada, for IPv4. LHR is now the busiest site, and although it is in Europe (although not in the European Union),

Use of the North American sites greatly lowers median latency. We show in Figure 10 the depolarization results for all sites combined, for IPv4 and IPv6. For both IPv4 and IPv6, we see median latency for all sites combined reducing 90 ms, from 100 to 10,ms. The IQR was reduced from 95 to 10 ms for IPv4. For IPv6, we observed few queries over TCP between Jan. 1 and 9, so they are not representative. After depolarizing, we see more queries over TCP. |

Op.	Day	Time	Prepend	Community	AMS(%)
1	21	15:00	2x	–	>0
2	22	9:53	2x	NE	>0
3	22	9:59	1x	–	100
4	22	10:21	1x	NE	100
5	22	10:37	1x	NE,15169:13000	100
6	22	11:00	2x	NE	>0

Table 6: BGP manipulations on AMS site of Anycast A – IPv4 and IPv6 prefixes to Google (AS15169) on Jan 21, 2020 (Time in UTC). NE: No Export

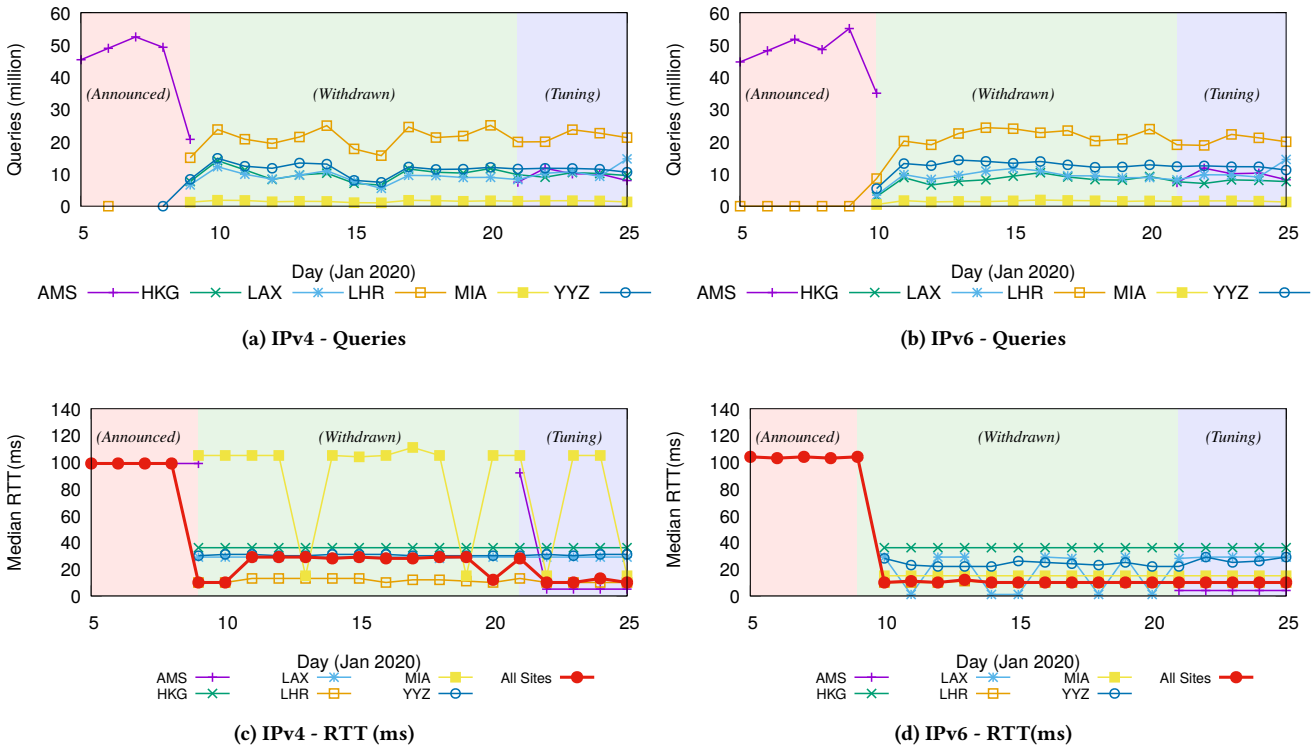
Although overall latency improves, omitting the AMS site misses the opportunity to provide better latency to their datacenters in the Netherlands and Denmark. We therefore resumed peering over the BGP session, with experimented with several policy routing choices shown in Table 6. We experimented with 1x and 2x AS-PATH prepending, no-export, and a Google-specific "try-not-to-use this path" community string. We found that no-export and the community string had no effect, perhaps because of the BGP session, and neither did single prepending. However double AS-PATH prepending left AMS with about 10% of the total traffic load. Full details of our experiments are in an appendix (§B.1).

4.3.3 Depolarizing Microsoft to `.nl` *Anycast A.* **Detection:** We discovered Microsoft anycast polarization through analysis of DNS/TCP across ASes. Microsoft's preferred site for `.nl` Anycast A is Miami (MIA), a different preference than Google's, but the outcome was the same: huge latency (median 80 ms) because global traffic goes to one place.

Resolution: Again, we worked with the operators at `.nl` Anycast A MIA and Microsoft to diagnose and resolve the problem. We confirm that Anycast had a peering session with Microsoft in MIA, and not at any other sites. Again, the result was a short AS-PATH and a preference for all Microsoft datacenters to use the Microsoft WAN to this site rather than other `.nl` Anycast A anycast sites.

Options that could mitigate this polarization include de-peering with Microsoft in MIA, peering with Microsoft at other sites, or possibly BGP-based traffic engineering. Because our ability to experiment with BGP was more limited at this site, and we could not start new peerings at other sites, the operator at MIA de-peered with Microsoft at our recommendation.

Figure 11 shows latency for this AS before and after our solution. Removing the direct peering addressed the problem, and Microsoft traffic is now distributed across all `.nl` Anycast A sites. As a result, the IQR falls from about 80 ms to 13 ms.



(a) IPv4 - Queries

(b) IPv6 - Queries

(c) IPv4 - RTT (ms)

(d) IPv6 - RTT(ms)

Figure 9: `.nl` Anycast A: queries and median RTT per site from Google (AS15169) – January 2020.

The median latency also falls by 70 ms, from 90 ms to 20 ms. Our technique identifies problems with polarization, and shows the dramatic improvement that results.

4.4 Detecting BGP Misconfiguration in Near Real-Time

Because it poses no additional cost on the network, passive measurement of anycast latency with DNS/TCP is an ideal method for *continuous, on-the-fly* detection of BGP misconfiguration. We are currently using this analysis operationally at `.nl`. We next illustrate this use-case with one example from that deployment.

On 2020-04-08, DNS/TCP real-time monitoring detected a jump in median DNS RTT from 55 ms to more than 200 ms (see Figure 12) but only for IPv4 traffic, not IPv6.

To investigate this change, we evaluated the number of ASes (Figure 12), routers, and query rates (Figure 13). We see that all grew when latency fell: with many more ASes and about 3× more queries and resolvers. To rule out DDoS attacks or a sudden burst in popularity for our domain, we confirmed that these ASes and resolvers have migrated from other sites (mostly Germany, site FRA) and went to SYD. Since many of these clients are in Europe, this nearly antipodal detour explains the latency increase.

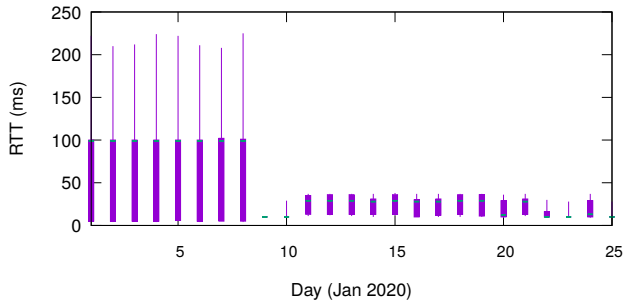
We reached out to the operator of `.nl` Anycast B SYD. They were confirmed and were already aware of the routing change. They informed us that a set of their SYD prefixes had accidentally to propagated through a large, Tier-1 transit provider. Since this provider peered with many other ASes in many places around the global, their propagation of the Anycast B anycast prefix provided a shorter AS-Path and sent traffic to SYD.

We also confirmed these routing changes on the RIPE RIS database of routing changes [37]. (Details are in §B.2.)

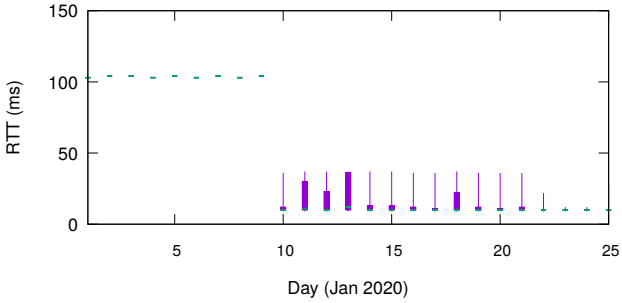
While catchment changes are not bad, route leaks that mis-route Europe to Australia are not an improvement. The lightweight nature of DNS/TCP observations of latency support 24x7 monitoring and allowed us to detect this problem. We are currently using it in operations at the `.nl`.

5 ANYCAST LATENCY EFFECTS ON TRAFFIC

While DNS/TCP can discover anycast latency, does latency matter? DNS caching means *users* are largely insulated from latency. However, we next confirm that latency does influence *traffic* to services when users have the choice of several. This effect was previously shown from clients [27], but not its impact on services.



(a) IPv4



(b) IPv6

Figure 10: Google depolarization results and RTT.

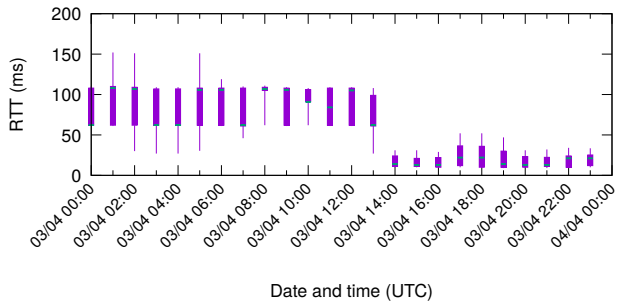


Figure 11: `.nl` Anycast B and Microsoft: RTT before and after depolarization.

Prior work has considered recursive resolver preference for lower latency [27]. Here we turn that analysis around and explore how changing anycast infrastructure shifts a client's preferences towards authoritative name servers. We confirm that lower latency results in increased traffic from recursive resolvers that have a choice between multiple anycast service addresses providing the same zone. (This question differs from studies that examine the optimality of a specific anycast service with multiple sites [19, 20].)

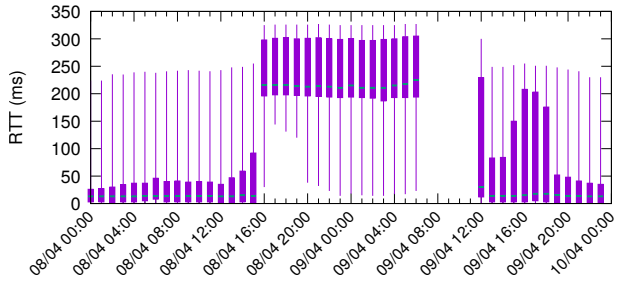


Figure 12: Anycast B SYD site: Latency for IPv4.

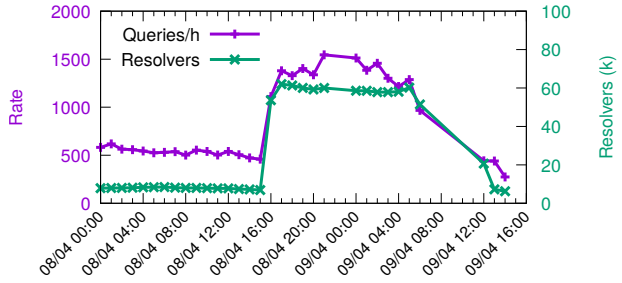


Figure 13: Anycast B SYD site: queries rate and resolvers.

To examine this question we use public RSSAC-002 statistics for the root server system [39]. From this we use the "traffic-volume" statistic, which reports queries per day for each root anycast service. (Recall that the Root DNS is provided by 13 different anycast service addresses per IP version, each using a different anycast infrastructure.) We show 6 months of data here (2019-11-01 to 2020-05-31), but we noticed similar trends since 2016. This analysis omits G- and I-Root, which did not provide during this period.

Figure 14 shows the fraction of traffic that goes to each anycast service in the root server system for one year. Two root letters deployed additional sites over this period: B-Root originally had 2 sites but added 3 sites in 2020-02-01, then optimized routing around 2020-04-01. H-Root originally had 2 sites but deployed 4 additional sites on 2020-02-11 and 3 additional sites on 2020-04-06. While other letters also added sites, B and H's changes were the largest improvements relative to their prior size. We see that, B and H's share rises from about 4% in 2019-11 to about 6% in 2020-05.

This data confirms that shows that when new sites are created at a root letter, they offer some clients lower latency for that letter. Lower latency causes some clients to shift more of their traffic to this letter, so its share of traffic relative to the others grows.

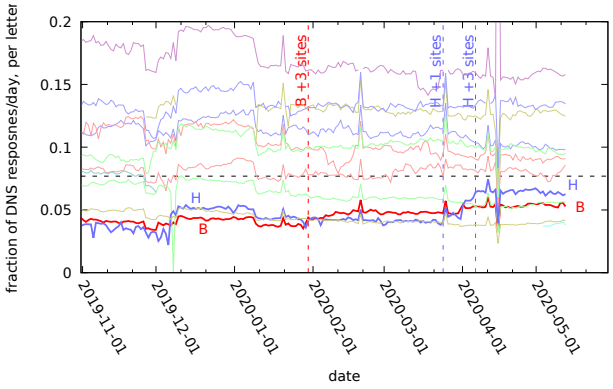


Figure 14: Fraction of traffic going to each root anycast service, per day, from RSSAC-002 data. B- and H-Root are bold lines.

6 RELATED WORK

Passive evaluation of TCP: Janey Hoe was the first to extract RTT from the TCP handshake [14], and it has been used by several groups since then (for example, in Facebook HTTP traffic [40]). We use this old idea, but are the first to apply it to DNS RTT estimation and to use to optimize Anycast DNS services.

Anycast performance and DNS: Anycast has been an active research topic over the last years. Ballani *et al.* [4] have proposed using a single upstream provider to avoid routing unexpected behavior. Schmidt *et al.* [42] have investigate the impact of *number of sites* and performance of anycast services, and pointed that sometimes, more sites may even lead to performance degradation. Moura *et al.* [24] have investigated how anycast react when DDoS attacks take place, by analyzing the 2015 attacks against the Root DNS servers [38]. They show how catchment affects how sites experience the query load distribution, with some sites becoming unavailable and others remaining active.

There are two general approaches to measure anycast latency today. First, RIPE Atlas [35] measures latency from about 11k physical devices distributed around the world. Verfploeter uses active probing to IPv4 to determine catchments [10, 11], and we have recent extended it to measure latency. Verfploeter provides data for about 5M /24 IPv4 networks. Our approach instead uses passive analysis of TCP traffic from real clients. It provides far better coverage than RIPE Atlas (§2.1). While Verfploeter provides coverage for millions of networks, our approach provides coverage for most of the networks that current generate traffic. In addition, since our analysis is passive, it places no additional strain on other networks and can run 24x7.

Li *et al.* [19] have proposed using new BGP communities to improve the site catchment, which, in turn, would requires protocol changes. Contrary to their approach, ours relies only on passive TCP traffic and does not involve protocol changes.

Anycast optimization for large CDNs with multiple providers: Going beyond how many sites and where to place them, McQuistin *et al.* [22] have investigated anycast networks with multiple upstream providers – which is typical for large CDNs. Given that each site may have its own set of peers/uspstreams, that may create inconsistencies, like in the case of Google and Anycast A (§4.3.2). They devise a methodology aims at daily measure the catchments of the network using active measurements, and compare where and how it changes for different configuration scenarios in terms of peers. Our work relates to them in the sense that we change peering for two sites of Anycast A in §4.3.2 and §4.3.3, in order to fix the polarization issue. Schlinker *et al.* [40] analyzes the change of catchments and use TCP RTT on Facebook's CDN, which most traffic is HTTP, and whose primary clients are real users. Authoritative DNS traffic, on the other hand, typically relates to the between resolver and authoritative server, and it's mostly UDP.

Performance-aware routing: Todd *et al.* [2] compare data from proposals for performance-aware routing from three content/cloud providers (Google, Facebook, and Microsoft) and show that BGP fares quite well for most cases. Others proposed to perform traffic engineering based on packet loss, latency and jitter [28, 32].

7 CONCLUSIONS

We have shown that DNS TCP connections are a useful source of latency information about anycast services for DNS. Although TCP is not (today) the dominant transport protocol for DNS, we showed that there is enough DNS/TCP to provide good coverage for latency estimation. We also showed how we prioritize use of this information to identify problems in operational anycast networks. We have used this approach to study three operational anycast services: two anycast servers of `.nl`, and one root DNS server (`B-root`). We documented one new class of latency problems: anycast polarization, an interaction where hypergiants get pessimal latency (100–200 ms) because of an poor interaction between their corporate backbones and global anycast services. We showed how we addressed this problem for `.nl`'s Anycast A with both Google and Microsoft. We also documented several other problems for anycast latency discovered through our analysis of DNS/TCP and showed that it enables continuous monitoring. We believe this approach will be of use to other DNS operators.

ACKNOWLEDGMENTS

We thank the operators of `.nl` Anycast A and B and `B-root` for their time and collaboration. We also thank Casey Deccio

for first proposing using TCP hanshake to measure DNS latency.

John Heidemann's research in this paper is supported in part by the DHS HSARPA Cyber Security Division via contract number HSHQDC-17-R-B0004-TTA.02-0006-I (PAADDOS), and by NWO.

Giovane C. M. Moura, Joao Ceron, Jeroen Bulten, and Cristian Hesselman research in this paper is supported by the Conconrdia Project, an European Union's Horizon 2020 Research and Innovation program under Grant Agreement No 830927.

REFERENCES

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. *DNS Security Introduction and Requirements.* RFC 4033. Internet Request For Comments. ftp://ftp.rfc-editor.org/in-notes/rfc4033.txt

[2] Todd Arnold, Matt Calder, Italo Cunha, Arpit Gupta, Harsha V. Madhyastha, Michael Schapira, and Ethan Katz-Bassett. 2019. Beating BGP is Harder than We Thought. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks* (Princeton, NJ, USA) *(HotNets '19)*. Association for Computing Machinery, New York, NY, USA, 9–16. https://doi.org/10.1145/3365609.3365865

[3] R. Austein. 2007. *DNS Name Server Identifier (NSID) Option.* RFC 5001. Internet Request For Comments. https://www.rfc-editor.org/rfc/rfc5001.txt

[4] Hitesh Ballani and Paul Francis. 2005. Towards a Global IP Anycast Service. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Philadelphia, Pennsylvania, USA) *(SIGCOMM '05)*. Association for Computing Machinery, New York, NY, USA, 301–312. https://doi.org/10.1145/1080091.1080127

[5] Hitesh Ballani, Paul Francis, and Sylvia Ratnasamy. 2006. A Measurement-based Deployment Proposal for IP Anycast. In *Proceedings of the 2006 ACM Conference on Internet Measurement Conference (IMC)*. ACM, 231–244.

[6] R. Bellis. 2010. *DNS Transport over TCP—Implementation Requirements.* RFC 5966. Internet Request For Comments. https://www.rfc-editor.org/rfc/rfc5966.txt

[7] Ray Bellis. 2015. Researching F-root Anycast Placement Using RIPE Atlas. ripe blog https://labs.ripe.net/Members/ray_bellis/researching-f-root-anycast-placement-using-ripe-atlas. https://labs.ripe.net/Members/ray_bellis/researching-f-root-anycast-placement-using-ripe-atlas

[8] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. 2013. Mapping the Expansion of Google's Serving Infrastructure. In *Proceedings of the ACM Internet Measurement Conference* (johnh: pafile). ACM, Barcelona, Spain, 313–326. https://doi.org/10.1145/2504730.2504754

[9] Sebastian Castro, Duane Wessels, Marina Fomenkov, and Kimberly Claffy. 2008. A Day at the Root of the Internet. *ACM Computer Communication Review* 38, 5 (April 2008), 41–46.

[10] Wouter B. de Vries, Salman Aljammaz, and Roland van Rijswijk-Deij. 2020. Global Scale Anycast Network Management with Verfploeter. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium* (johnh: pafile). IEEE, Budapest, Hungary.

[11] Wouter B. de Vries, Ricardo de O. Schmidt, Wes Haraker, John Heidemann, Pieter-Tjerk de Boer, and Aiko Pras. 2017. Verfploeter: Broad and Load-Aware Anycast Mapping. In *Proceedings of the ACM Internet Measurement Conference*. London, UK. https://doi.org/10.1145/3131365.3131371

[12] John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, and Bill Weihl. 2002. Globally Distributed Content Delivery. *IEEE Internet Computing* 6, 5 (Sept. 2002), 50–58. https://doi.org/10.1109/MIC.2002.1036038

[13] Google. 2020. Google BGP communities. https://support.google.com/interconnect/answer/9664829?hl=en. (Jan. 2020).

[14] Janey C. Hoe. 1996. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. In *Proceedings of the ACM SIGCOMM Conference* (johnh: pafiles). ACM, Stanford, CA, 270–280.

[15] P. Hoffman, A. Sullivan, and K. Fujiwara. 2018. *DNS Terminology.* RFC 8499. IETF. http://tools.ietf.org/rfc/rfc8499.txt

[16] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. 2016. *Specification for DNS over Transport Layer Security (TLS).* RFC 7858. Internet Request For Comments. https://doi.org/10.17487/RFC7858

[17] ICANN. 2014. RSSAC002: RSSAC Advisory on Measurements of the Root Server System. https://www.icann.org/en/system/files/files/rssac-002-measurements-root-20nov14-en.pdf.

[18] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. 2010. Internet Inter-Domain Traffic. In *Proceedings of the ACM SIGCOMM Conference* (johnh: pafile). ACM, New Delhi, India, 75–86. https://doi.org/10.1145/1851182.1851194

[19] Zhihao Li, Dave Levin, Neil Spring, and Bobby Bhattacharjee. 2018. Internet Anycast: Performance, Problems, & Potential. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (Budapest, Hungary) *(SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 59–73. https://doi.org/10.1145/3230543.3230547

[20] Jinjin Liang, Jian Jiang, Haixin Duan, Kang Li, and Jianping Wu. 2013. Measuring Query Latency of Top Level DNS Servers. In *Proceedings of the International conference on Passive and Active Measurements (PAM)*. 145–154.

[21] D. McPherson, D. Oran, D. Thaler, and E. Osterweil. 2014. *Architectural Considerations of IP Anycast.* RFC 7094. IETF. http://tools.ietf.org/rfc/rfc7094.txt

[22] Stephen McQuistin, Sree Priyanka Uppu, and Marcel Flores. 2019. Taming Anycast in the Wild Internet. In *Proceedings of the Internet Measurement Conference* (Amsterdam, Netherlands) *(IMC '19)*. Association for Computing Machinery, New York, NY, USA, 165–178. https://doi.org/10.1145/3355369.3355573

[23] P.V. Mockapetris. 1987. *Domain names - implementation and specification.* RFC 1035. IETF. http://tools.ietf.org/rfc/rfc1035.txt

[24] Giovane C. M. Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Müller, Lan Wei, and Christian Hesselman. 2016. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In *Proceedings of the ACM Internet Measurement Conference* (johnh: pafile). ACM, Santa Monica, California, USA, 255–270. https://doi.org/10.1145/2987443.2987446

[25] Giovane C. M. Moura, John Heidemann, Ricardo de O. Schmidt, and Wes Hardaker. 2019. Cache Me If You Can: Effects of DNS Time-to-Live (extended). In *Proceedings of the ACM Internet Measurement Conference* (johnh: pafile). ACM, Amsterdam, the Netherlands, to appear. https://doi.org/10.1145/3355369.3355568

[26] Moritz Müller, Giovane C. M. Moura, Ricardo de O. Schmidt, and John Heidemann. 2017. Recursives in the Wild: Engineering Authoritative DNS Servers. In *Proceedings of the ACM Internet Measurement Conference* (johnh: pafile). ACM, London, UK, 489–495. https://doi.org/10.1145/3131365.3131366

[27] Moritz Müller, Giovane C. M. Moura, Ricardo de O. Schmidt, and John Heidemann. 2017. *Recursives in the Wild: Engineering Authoritative DNS Servers.* Technical Report ISI-TR-720. USC/Information Sciences Institute.

[28] Priyadarsi Nanda and AJ Simmonds. 2009. A scalable architecture supporting QoS guarantees using traffic engineering and policy based routing in the internet. *International Journal of Communications, Network and System Sciences* (2009).

[29] T. Narten, R. Draves, and S. Krishnan. 2007. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.* RFC 4941. Internet Request For Comments. ftp://ftp.rfc-editor.org/in-notes/rfc4941.txt

[30] C. Partridge, T. Mendez, and W. Milliken. 1993. *Host Anycasting Service.* RFC 1546. IETF. http://tools.ietf.org/rfc/rfc1546.txt

[31] Enric Pujol, Ingmar Poese, Johannes Zerwas, Georgios Smaragdakis, and Anja Feldmann. 2019. Steering hyper-giants' traffic at scale. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*. 82–95.

[32] Bruno Quoitin, Cristel Pelsser, Olivier Bonaventure, and Steve Uhlig. 2005. A performance evaluation of BGP-based traffic engineering. *International journal of network management* 15, 3 (2005), 177–191.

[33] RIPE NCC. 2019. RIPE Atlas Measurement IDS. https://atlas.ripe.net/measurements/ID. , where ID is the experiment ID: DNS/TCP:22034303, DNS/UDP: 22034324, GoDNS-21: 23859473 , GoTrace-21: 23859475 GoDNS-22: 23863904, GoTrace-22: 23863901, ComcastV6: 24269572, ComcastV6-afterReport: 24867517, ChinaNetV6: 24257938.

[34] RIPE NCC. 2020. RIPE Atlas Probes. https://ftp.ripe.net/ripe/atlas/probes/archive/2020/05/.

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

[36] RIPE Network Coordination Centre. 2015. RIPE Atlas. https://atlas.ripe.net.

[37] RIPE Network Coordination Centre. 2020. RIPE - Routing Information Service (RIS). https://https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris.

[38] Root Server Operators. 2015. Events of 2015-11-30. http://root-servers.org/news/events-of-20151130.txt.

[39] Root Server Operators. 2019. Root DNS. http://root-servers.org/.

[40] Brandon Schlinker, Italo Cunha, Yi-Ching Chiu, Srikanth Sundaresan, and Ethan Katz-Bassett. 2019. Internet Performance from Facebook's Edge. In *Proceedings of the Internet Measurement Conference* (Amsterdam, Netherlands) *(IMC '19)*. ACM, New York, NY, USA, 179–194. https://doi.org/10.1145/3355369.3355567

[41] Brandon Schlinker, Hyojeong Kim, Timothy Cui, Ethan Katz-Bassett, Harsha V. Madhyastha, Italo Cunha, James Quinn, Saif Hasan, Petr Lapukhov, and Hongyi Zeng. 2017. Engineering Egress with Edge Fabric: Steering Oceans of Content to the World. In *Proceedings of the ACM SIGCOMM Conference* (johnh: pafile). ACM, Los Angeles, CA, USA, 418–431. https://doi.org/10.1145/3098822.3098853

[42] Ricardo de O. Schmidt, John Heidemann, and Jan Harm Kuipers. 2017. Anycast Latency: How Many Sites Are Enough?. In *Proceedings of the Passive and Active Measurement Workshop* (johnh: pafile). Springer, Sydney, Australia, 188–200. http://www.isi.edu/%7ejohnh/PAPERS/Schmidt17a.html

[43] SIDN Labs. 2020. ENTRADA - DNS Big Data Analytics. https://entrada.sidnlabs.nl/.

[44] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, and Bruce Maggs. 2014. The Internet at the Speed of Light. In *Proceedings of the* (johnh: pafile). ACM, Los Angeles, CA, USA. https://doi.org/10.1145/2670518.2673876

[45] Steve Souders. 2008. High-Performance Web Sites. *Commun. ACM* 51, 12 (Dec. 2008), 36–41. https://doi.org/10.1145/1409360.1409374

[46] Paul Vixie. 2012. Response Rate Limiting in the Domain Name System (DNS RRL). blog post http://www.redbarn.org/dns/ratelimits. http://www.redbarn.org/dns/ratelimits

[47] Duane Wessels. 2020. RSSAC002-data. https://github.com/rssac-caucus/RSSAC002-data/.

[48] Maarten Wullink, Giovane CM Moura, Moritz Müller, and Cristian Hesselman. 2016. ENTRADA: A high-performance network traffic data streaming warehouse. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 913–918.

[49] Pengxiong Zhu, Keyu Man, Zhongjie Wang, Zhiyun Qian, Roya Ensafi, J. Alex Halderman, and Haixin Duan. 2020. Characterizing Transnational Internet Performanceand the Great Bottleneck of China. In *Proceedings of the ACM SIGMETRICS conference*. ACM, Boston, MA, USA, *(to appear)*.

ASN	CC	Owner
27717	VE	Corporacion Digitel C.A.
4837	CN	CHINA169-BACKBONE CNCGROUP China169 Backbone
38266	IN	VODAFONE-IN Vodafone India Ltd.
20940	EU	AKAMAI-ASN1
45899	VN	VNPT-AS-VN VNPT Corp
6057	UY	Administracion Nacional de Telecomunicaciones
10620	CO	Telmex Colombia S.A.
28573	BR	CLARO S.A.
14061	US	DIGITALOCEAN-ASN - DigitalOcean, LLC
1267	EU	ASN-WIND IUNET
45356	LK	MOBITEL-LK IS Group, No:108, W A D Ramanayake Mawatha
7629	PH	EPLDT-AS-AP 5F L.V. Locsin Bldg
32934	US	FACEBOOK - Facebook, Inc.
10429	BR	Telefonica Data S.A.
4007	NP	SUBISU-CABLENET-AS-AP Subisu Cablenet (Pvt) Ltd, Baluwatar, Kathmandu, Nepal
9299	PH	IPG-AS-AP Philippine Long Distance Telephone Company
35819	SA	MOBILY-AS Etihad Etisalat Company (Mobily)
6147	PE	Telefonica del Peru S.A.A.
18881	BR	TELEFÔNICA BRASIL S.A
47211	RU	KOLPINONET-AS

Table 7: Top 20 talker organizations to `B-root`– see Figure 5.

A EXTRA `B-ROOT` DATA

Table 7 shows all top talkers by organizastion, and Table 8 by IPv4, and Table 9 by IPv6.

Figure 15 shows additiona latency information.

B CASE STUDY DETAILS

B.1 Details of the `.nl` Anycast A AMS site Peering with Google

Bringing back AMS. We first start by announcing our prefixes with 2x prepending towards Google in AMS. That causes *part* of the traffic to AMS (Figure 16), as we intended. Adding No-export option to the announcement, on the 22nd did not have any effect (private peering) We experimented further (operations 3–4) with prepeding our announcement only 1 time. That, in turn, caused Google to become polarized again for Anycast A

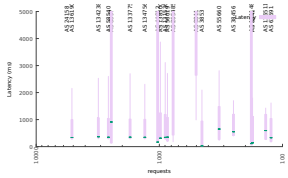
We also carried measurements from 20 Atlas probes located at Google' s networks during these changes, as shown in Table 10. Figure 17 show these results for Altas.

So we determined empirically that 2x prepend solved the issue, and 1x did not. We wondered if we could reduce the polarization with BGP communities while keeping prepending to 1x. We tested at Op. 5 the the community `15169:13000`, which is provided by Google places lowest priority on BGP

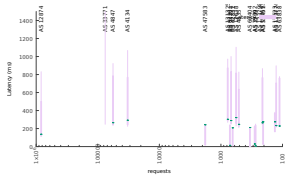
ASN	CC	Owner
9811	CN	BJGY srit corp.,beijing.
24158	TW	TAIWANMOBILE-AS Taiwan Mobile Co., Ltd.
44812	RU	IPSERVER-RU-NET Fiord
265485	BR	UP NET TELECOM
131148	TW	BOT-AS-TW Bank Of Taiwan
8897	GB	KCOM-SPN (Service-Provider Network) (ex-Mistral)
55660	ID	MWN-AS-ID PT Master Web Network
138500	NP	AS138500
136511	PH	BEC-AS-AP Broadband Everywhere Corporation
3853	US	WHIDBEY - Whidbey Telephone Company
134756	CN	CHINANET-NANJING-IDC CHINANET Nanjing IDC network
58540	CN	CHINATELECOM-HUNAN-ZHUZHOU-MAN Zhuzhou
134238	CN	CT-JIANGXI-IDC CHINANET Jiangx province IDC network
38456	AU	SPEEDCAST-AU SPEEDCAST AUSTRALIA PTY LIMITED
133775	CN	CHINATELECOM-FUJIAN-XIAMEN-IDC1 Xiamen
4913	US	NET-CPRK - Harris CapRock Communications, Inc.
5377	NO	MARLINK-EMEA
55722	NR	CENPAC-AS-AP Cenpac Net Inc
134771	CN	CHINATELECOM-ZHEJIANG-WENZHOU-IDC WENZHOU, ZHEJIANG Province, P.R.China.
136190	CN	CHINATELECOM-YUNNAN-DALI-MAN DaLi

Table 8: Top 20 IPv4 ASNs with large handshake latency distributions.

-



(a) `B-root` AS latency diversity for IPv4 by data size



(b) `B-root` AS latency diversity for IPv6 by datasize

Figure 15: Latency analysis to `B-root` by AS latency diversity.

ASN	CC	Owner
133798	ID	SMARTFREN-AS-ID PT. Smartfren Telecom, Tbk
33771	KE	SAFARICOM-LIMITED
23724	CN	CHINANET-IDC-BJ-AP IDC, China Telecommunications Corporation
63838	CN	CT-HUNAN-HENGYANG-IDC Hengyang
12874	IT	FASTWEB
55451	TH	AS55451
47583	LT	AS-HOSTINGER
60404	NL	LITESERVER
17623	CN	CNCGROUP-SZ China Unicom Shenzen network
42831	GB	UKSERVERS-AS UK Dedicated Servers, Hosting and Co-Location
4809	CN	CHINATELECOM-CORE-WAN-CN2 China Telecom Next Generation Carrier Network
17754	IN	EXCELL-AS Excellmedia
202196	NL	BOOKING-BV
12989	NL	HWNG
264496	BR	IR TECNOLOGIA LTDA ME
37009	NA	MTCASN
7633	IN	SOFTNET-AS-AP Software Technology Parks of India - Bangalore
45899	VN	VNPT-AS-VN VNPT Corp
27435	US	OPSOURCE-INC - Dimension Data Cloud Solutions, Inc.
45570	AU	NETPRES-AS-AP Network Presence

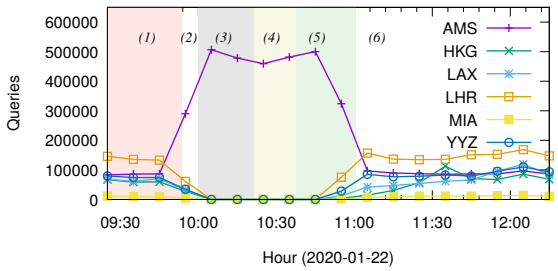
Table 9: Top 20 IPv6 ASNs with large handshake latency distributions.

ID	Type	Frequency
GoDNS-21	DNS `hostname.bind`	300s
GoTrace-21	Traceroute	300s
GoDNS-22	DNS `hostname.bind`	300s
GoTrace-22	Traceroute	900s

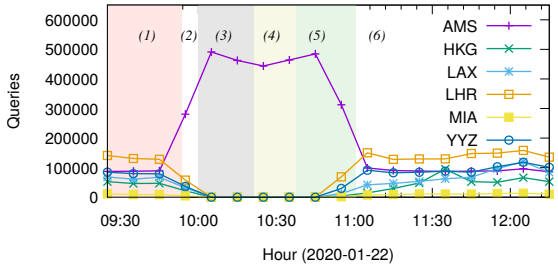
Table 10: Ripe Atlas measurements used in the AS15169 tuning experiments on (2020-01-21 and 22) [33].

choices ("try to not serve traffic here") [13]. We see in Figure 18 that this operation (10:37) caused not catchment changes, and, as such, the communities did not influence traffic selection. Even though Google supports BGP communities, the ultimate decision on how to use it its own their side (and they are clear about it [13]). This particular example showcase when communities do not work as one hoped for.

Last, we prepend the announcement 2x, and traffic resume going to AMS. As can be seen in Figure 9, the median RTT for Google and Anycast A improved significantly. Overall, BGP operations do not always work as one would expect.

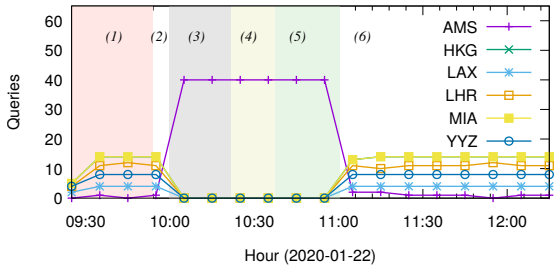


(a) IPv4 - Anycast A



(b) IPv6 - Anycast A

Figure 16: Catchment changes according to BGP manipulations (from Anycast A passive data).



(a) IPv4 - Ripe Atlas 20 VPs from Google AS15169

Figure 17: Catchment changes according to BGP manipulations from Ripe Atlas.

B.2 Details about Detections of BGP Misconfiguration

By investigating routing events associated with Anycast B on RIPE RIS [37] we could confirm changes in the route visibility of the results reported in §4.4.. RIPE RIS is a public database that collects and store Internet routing data in several points distributed globally, typically located in IXPs. Internet data routing data consist of BGP messages observed by collectors, including prefix announcement, prefix withdraw and update

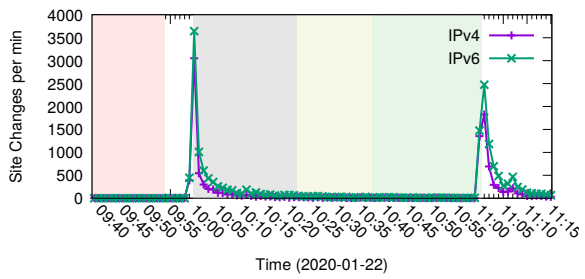


Figure 18: Timeseries (per minute) of Google' s resolver changing sites on Anycast A.

messages. For instance, it is possible to have a historical view of the paths to a specific destination.

Figure 19 shows the number of paths to Anycast B seen by Route RIS collectors. On 2020-04-08 at 15:59:50 we observed several BGP update messages that have triggered an increase of the number of paths to Anycast B, from 340 to 362. The visibility of Sever B became quite steady till the next day, on 2020-04-08 at 06:37, when, after a few BGP withdraw messages, it has returned again to 340.

The routing events coincide with the observation of our DNS/TCP RTT measurement as depicted on Figure 12. Routing updates force BGP neighbors to reprocess the routing table and update the best route selection. Thus, after new routing paths provided by the first set of BGP update messages (2020-04-08 at 15:59:50) the traffic was shifted to new routes and ended up in our site in SYD. The situation changed back to the previous situation when the new routing path was removed on 2020-04-08 at 06:37.

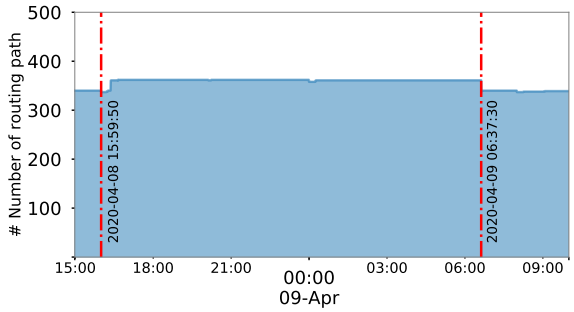


Figure 19: Number of paths to Sever B seen by Route RIS collectors.