An Empirical Evaluation of Longitudinal Anycast Catchment Stability

Remi Hendriks*, Bernhard Degen*, Bas Palinckx, Raffaele Sommese, and Roland van Rijswijk-Deij

University of Twente, Enschede, the Netherlands

Abstract. Anycast is widely used to improve the availability and performance of, e.g., the DNS and content delivery. To use anycast one simply announces the same prefix from multiple points of presence (PoPs). BGP then takes care of routing clients to the "nearest" PoP. While seemingly simple, managing an anycast service is not without challenges. Factors outside operator control, such as remote peering and hidden MPLS tunnels, may route clients to suboptimal PoPs in terms of latency.
To successfully manage anycast, operators map catchments: the set of prefixes that reach each PoP of a service. Due to the dynamic nature of inter-domain routing, catchments change over time. While earlier work has looked at catchment stability in the short term and in a coarse-grained manner, we lack a detailed view on catchment stability over time. Understanding this is crucial for operators as it helps schedule catchment measurements and plan interventions using traffic engineering to redistribute traffic over PoPs.
In this work, we put long-term catchment stability on an empirical footing. Using an industry-grade anycast testbed with 32 globally distributed PoPs, we continuously map catchments for both IPv4 and IPv6 over a 6-month period and study catchment stability at different timescales, ranging from days to weeks and months. We show catchments are very stable in the short term, with 95% of prefixes routing to the same PoP on a one-week scale, and over 99% on a day-by-day basis. We also show, however, that sudden routing changes can have a major impact on catchments. Based on our longitudinal results, we make recommendations on the frequency with which to measure catchments.

1 Introduction

Anycast, a technique to use the same host address at multiple locations [10], is widely used to improve the resilience, availability and performance of services. For example, the DNS root system makes use of anycast to ensure that root server instances are highly available, close to end users [13]. Similarly, content delivery networks (CDNs) make use of anycast to steer clients to nearby replicas of popular content, enhancing the user experience by lowering latency and distributing the load of client populations over multiple replicas (e.g., [14]).

* Shared first author

While the basic principle of anycast – using the same host address at multiple locations – is simple, managing an anycast service is challenging. Factors outside the control of an operator, ranging from remote peering, hidden tunnels and routing changes by upstreams, can adversely affect the performance of an anycast service, leading to clients ending up at a Point of Presence (PoP) that is suboptimal in terms of latency. To this end anycast operators make use of catchment mapping: determining which client prefixes end up at which PoP. Suboptimal mappings can then be addressed through traffic engineering, as discussed by, e.g., Koch et al. [8]. As we will discuss in §2, different methods exist for mapping anycast catchments. What all prior studies have in common, however, is that none study anycast catchment stability over significant periods of time. Understanding this is crucial for two reasons. First, operators need to make informed choices on how often to reassess anycast catchments, allowing them to plan when to update traffic engineering policies, or, e.g., to plan PoP maintenance windows. Second, work by De Vries et al. [17] has shown that accurate information about a PoP's catchment can help filter out spoofed denial-of-service traffic that ingresses at that PoP.

In this work, we therefore set out to put anycast catchment stability on an empirical footing. Using an industry-grade testbed consisting of 32 globally distributed nodes we perform daily catchment mappings for IPv4 and IPv6 using the state-of-the-art "Verfploeter" approach pioneered by De Vries [18]. We use the data we collect to study long-term anycast catchment stability. Our results show that in the short term, anycast catchments are very stable, and even in the long term, catchments remain relatively stable. We also show, however, that catchments may change abruptly when major routing events occur and that unannounced interventions by our commercial upstream can lead to drastic traffic shifts. We use our results to make recommendations for operators on how frequently to schedule catchment measurements.

2 Background and Related Work

Anycast – Anycast was first introduced in 1993 [10]. The goal of anycast is to distribute a service over multiple servers that share the same host address. Each site of an anycast service – usually referred to as a Point of Presence (PoP) – then announces the prefix that contains the host address(es) of the anycast service via BGP. The inter-domain routing system then routes clients to the "nearest" server in terms of topology and routing policy. Anycast is used extensively in the DNS (e.g., by the root servers) and by Content Delivery Networks (CDNs).

Catchment mapping – as discussed, to manage an anycast service operators typically need to know which clients reach which PoP. In other words: clients from which parts of the address space are routed to which PoP. This information is useful for dimensioning PoPs in terms of resources, troubleshooting suboptimal routing, where clients get routed to remote PoPs with higher latency while a physically closer PoP is available, and understanding which PoPs clients switch to when a PoP is taken down for maintenance or hit by an outage.

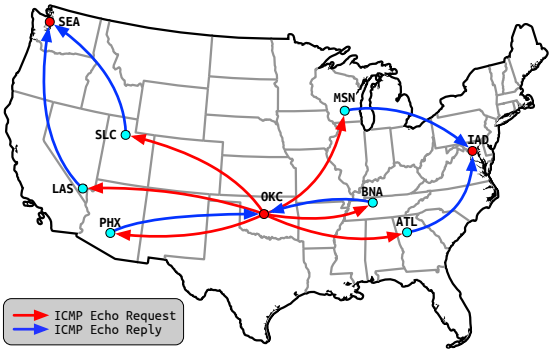


Fig. 1: The Verfploeter [18] method; an anycast site (red) sends ICMP probes to a hitlist, responses from clients (blue) end up at the "closest" anycast site.

Earlier studies have introduced (combinations of) methods to map anycast catchments. A common method is to use external vantage points, either public vantage points, such as RIPE Atlas or PlanetLab [2,3,4,8,9,14], or openly available systems, such as open DNS resolvers [4]. A limitation of this approach is that these external vantage points typically represent only a fraction of the address space. An alternative is to analyze log data of anycast PoPs [2,6,8,14], but this only works if a service is already active and cannot be used for exploratory mapping to, e.g., optimise PoP placement before launching a service. Client-side measurements, e.g., injecting measurement code in served web pages [2], offer an attractive solution as they represent real users, but this approach is limited to the Web. Finally, Verfploeter – which we explain in detail below – uses active probing from an anycast address as source to map catchments based on probe responses from a hitlist of clients [18]. This approach has the advantage that it can be used independent of a running anycast service, and can target a wide range of clients. A limitation is that clients need to be responsive to probes.

Verfploeter – In this paper we make use of the Verfploeter anycast catchment mapping technique, as it is easy to deploy and does not require us to run an active anycast service. Figure 1 provides a bird's eye overview of how the technique works. The intuition is as follows: the anycast service (red circles in the figure) sends out ICMP Echo Request probes to a hitlist of IP addresses (e.g., the USC ISI hitlist [16]) and uses an anycasted host address as source address (red arrows). Responsive hosts (blue circles) on the hitlist will then respond to this anycast address, and will automatically be routed to the "closest" PoP of the anycast services (blue arrows). This allows the anycast operator to map the catchment of each PoP.

Catchment stability – so far, we have discussed methods to measure anycast catchment. What we are interested in for this paper is the longitudinal stability of these catchments, in other words: how often and to what extent do catchments of PoPs change over shorter or longer periods of time? Previous works [2,17,18] have looked at catchment stability but only in the short term, ranging from a single

day to a maximum of two weeks. In this paper, we bridge this gap based on an in-depth study of stability using day-to-day measurements spanning 6 months.

3 Methodology and Dataset

Methodology – Our measurement setup consists of an anycast testbed comprising 32 nodes, geographically distributed across 20 countries on six continents [1]. The nodes are hosted on Vultr, one of the few cloud providers that supports BGP announcements for smaller customers. We argue that our testbed is representative of small to medium sized industry-grade anycast deployments, as Vultr is widely used for anycast services. Based on a recent measurement of anycast services we obtained from the MAnycast2 team [15], we see 130 ASNs using Vultr to anycast prefixes, of which 33 exclusively use Vultr. Users include, e.g., top-level domain operators, such as SIDN for `.nl` and TWNIC for `.tw`. We note that the effectiveness of our method may require further investigation for larger-scale networks such as Cloudflare with hundreds of nodes.

We have been actively announcing a /24 IPv4 prefix and a /40 IPv6 prefix at all sites since March 21, 2024. While there have been occasional instances of site inactivity due to Vultr outages or technical issues, such occurrences are rare, and the deployment has remained largely consistent over time.

Each day at 00:00h UTC, we perform a catchment mapping using the Verfploeter methodology. To determine the catchment targets (i.e., the clients for which we wish to establish their corresponding anycast node catchment), we utilize several hitlists. For IPv4, we employ the hitlist provided by USC/ISI as part of their IPv4 Internet census [16]. For IPv6, we use a combination of sources: the TU Munich IPv6 hitlist [5] and a list of IPv6 addresses taken from `AAAA` DNS records collected by the OpenINTEL project [12]. These sources tend to fluctuate over time as we update them; therefore, in our analyses we only consider prefixes that are present throughout the entire period of our study.

We perform our measurements using ICMP. To ensure that our findings hold for transport-layer protocols, we conducted a scan using TCP. We find 1.1 M prefixes responsive to both TCP and ICMP, of these 97.7% have a consistent catchment. We suspect that the remaining 2.3% route differently due to *e.g.,* route flips and load-balancing. Overall, these results show that our catchment mappings with ICMP hold for transport-layer protocols. Furthermore, analysis shows that we do not significantly increase coverage by considering both which is why we only consider catchment mappings using ICMP.

We also augment the original Verfploeter concept by extending sending of probes from a single PoP to distributed sending from all PoPs. This improves our ability to run faster back-to-back scans, with less packet loss. We verified that this changed approach produces accurate mappings that match mappings where probes are sent from a single anycast node. Our daily measurements generate approximately 1,000 packets per second from all sites, with a completion time of 95 minutes. To investigate sub-daily variations in catchment behavior, we scheduled an hourly measurement from a single node in Frankfurt from Septem-

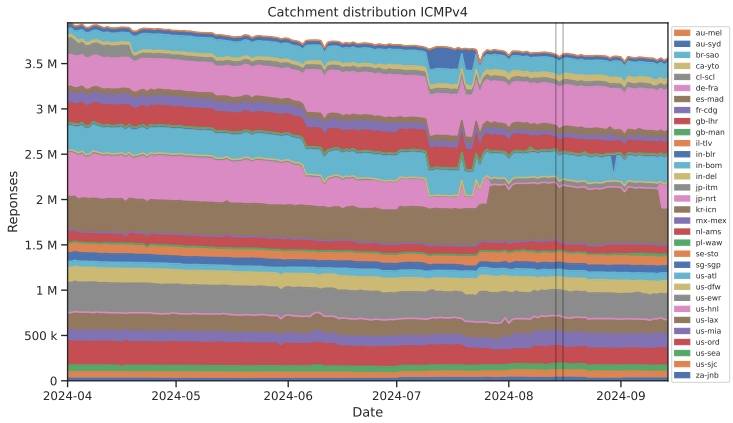


Fig. 2: Distribution of /24 prefixes over catchments. At the start there are about 4 M responsive /24 prefixes. The hitlist was updated on March 23 (before the start of our analyses), and August 14 and 16 (indicated by vertical lines). Catchments size across different nodes highly varies due to geographical interconnections. The responsiveness decline is mostly attributable to the hitlist aging.

ber 17, 2024, at a higher rate of 5,000 packets per second to ensure full hitlist scanning completes within one hour.

Ethics – in our measurement traffic, we encode the sending site, receiving site, transmission time, and reception time in the ICMP payload, along with an opt-out URL and a description of the measurements. As part of our responsible scanning practices, we have updated the rDNS records and marked the abuse contacts in the RIR data associated with our testbed prefixes to facilitate easy opt-out for operators.

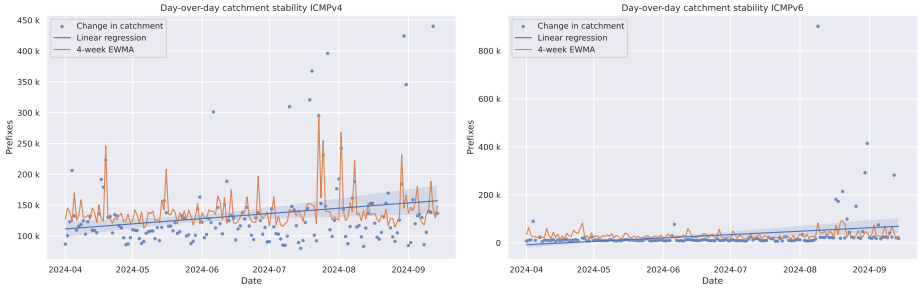
Dataset – in total, we collected an average of 3.7 M responsive IPv4 /24 prefixes and 0.7 M responsive IPv6 /40 prefixes over 178 days between March 21, 2024, and September 14, 2024. Of these, 3.5 M IPv4 /24 prefixes and 0.5 M IPv6 /40 prefixes were consistently responsive throughout the entire analysis period.

4 Results

In this section, we present the results of our analyses, focusing first on the stability of catchments over time, and then investigating catchment changes in more detail by presenting a case study on a specific anycast node that experienced a significant shift in its catchment.

4.1 Catchment stability over time

As a first step in our analysis, we evaluated the catchment size over time starting from April 1st, 2024 for each node. We chose to begin our analysis on April

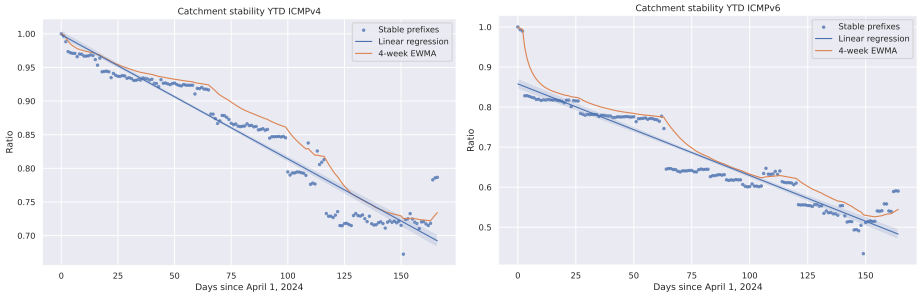


(a) ICMPv4 (b) ICMPv6

Fig. 3: Stability of catchments day-over-day. The points are the number of prefixes falling in different catchments as compared to the previous day. The blue line is a linear regression model fit to the stability data. The translucent bands depict the corresponding confidence interval at level $\gamma = 95\%$. The orange line is the exponentially-weighted average over the last 28 days.

1st, rather than March 21st, to avoid the instabilities observed during the initial days following the announcement. Figure 2 shows the catchment distribution over time for all 32 nodes in our testbed. The first notable observation is the significant variance in catchment size across different nodes for IPv4 addresses. This variation is largely attributable to the diverse geographical distribution of the nodes in our testbed, the size of the regions they serve in terms of IP addresses, and the networks to which the nodes are interconnected. Another point of interest is that at the beginning of our measurements, we observe approximately 4 million responses; however, as the hitlist ages, we lose approximately 0.5 million responses. This result is unsurprising, and is mostly attributable to the hitlists aging as the authors of the ISI hitlist also observed this behavior in their original study [11]. Figure 2 also highlights several notable events. On March 27, kr-icn's catchment more than triples in size, while us-sea and jp-nrt experience a reduction. On July 10, jp-nrt loses the majority of its traffic, while au-syd's catchment expands by nearly five times. Between July 27 and September 12, kr-icn observes approximately 200 k more prefixes than usual, while jp-nrt's catchment shrinks. Additionally, on August 30, in-blr experiences a short-lived growth in catchment. Later in the paper (Section 4.2), we will investigate some of these events to understand the reasons behind these catchment shifts.

The catchment sizes illustrated above provide insight into the degree of hegemony a particular node exhibits over time across a series of clients. However, they do not offer a clear picture of how many prefixes shift from one catchment to another. To address this, we examined the day-to-day churn in anycast catchments by plotting the number of prefixes that shift between catchments. In Figure 3, we observe that while most daily catchment shifts for all nodes range between 100 k and 150 k prefixes, there are occasional spikes, likely caused by rerouting events, that lead to significant changes in catchment size. This behavior is clearly visible in both IPv4 (Figure 3a) and IPv6 (Figure 3b), with spikes reaching up



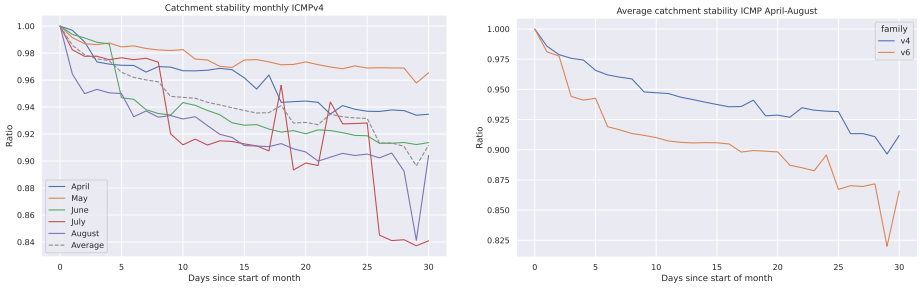
(a) ICMPv4 (b) ICMPv6

Fig. 4: Stability of catchments since the first measurement. The ratio of stable prefixes at time t_x is calculated as the number of prefixes in the same catchment at times t_0 and t_x divided by the number of prefixes responsive at times t_0 and t_x. The blue line is a linear regression model fit to the stability data. The translucent bands depict the corresponding confidence interval at level $\gamma = 95\%$. The orange line is the exponentially-weighted average of the last 28 days.

to 400 k prefixes shifting catchments in a single day. For IPv6, we also observed a shift of 900 k prefixes in August. After deeper investigation, we attributed this anomaly to a significant inflation of the IPv6 hitlist we use on that day. We also investigated sub-daily changes, by mapping the catchment hourly. In this case, we saw ∼78 k ($< 2\%$ of the total) prefixes shifting catchment every hour compared to 4 M that remain stable. Overall, the results indicate that anycast catchments remain relatively stable in day-to-day operations.

While catchments remain stable on a day-to-day basis, we want to understand whether this stability holds over a more extended period. In other words, *to what extent does the catchment measured at a specific point reflect the future state of an anycast deployment's catchment?* To explore this aspect, we evaluated the stability of catchments over time by focusing only on the responsive prefixes throughout the entire analysis period. We considered the catchment on the first day of our analysis, April 1$^{\text{st}}$, as the representative catchment and calculated how much each subsequent day's catchment differed from that of the first day. This was done by computing the ratio between the number of prefixes remaining in the same catchment on a given day and the number of prefixes in that catchment on the first day. A ratio of 1 indicates that all prefixes remain in their original catchment, while, for example, a ratio of 0.5 indicates that half of the prefixes have moved to the catchment of other anycast nodes.

For IPv4 (Figure 4a), we observe that the original mapping remains valid for up to 2 months if operators accept an error rate (i.e., % of prefixes wrongly mapped) of 10% of the original catchment. We also notice that the decline is fairly gradual on average (based on a 4-week moving average), though there are days where this accelerates, likely due to re-routing events. For IPv6 (Figure 4b), we observe a more turbulent behavior. In this case, after just 4 days, the original catchment mapping no longer reflects the current status, with nearly 20% of the



(a) Monthly stability with respect to the beginning of the month.

(b) Average monthly stability of ICMPv4 and ICMPv6 compared. We observe that IPv4 routing appears more stable on average and exhibits less variability.

Fig. 5: Stability of catchments over time. The ratio of stable prefixes at time t_x is calculated as the number of prefixes in the same catchment at times t_0 and t_x divided by the number of prefixes responsive at both times t_0 and t_x.

clients shifting to a different node. Upon deeper investigation, we attribute this behavior to two primary factors: the smaller size of the IPv6 catchment, which is 7-fold smaller than the respective IPv4 catchment, leading to an amplification of shifting trends in our analysis, and the extreme variability of the IPv6 hitlist we use to map catchments, as also noted by its creators [5].

The previous analysis heavily depends on the representativeness of the selected starting day. To verify whether the previous assumptions hold true when selecting different starting days as the representative catchment, we computed the same catchment stability for different months, using the first day of each month as the representative day. As seen in Figure 5a, while catchments remain stable on average until the end of the month, in two of the worst-case scenarios (July and August), the representativeness of the original catchment deteriorates rapidly after the 3$^{\text{rd}}$ week. We therefore suggest that a scan every 2-3 weeks is the minimum requirement for a reasonable indication of the catchment for IPv4.

In contrast, for IPv6, we observe a faster decline on average across all the months analyzed. This decline, in addition to the fast turnover of the IPv6 hitlist, suggests that more frequent scans of the catchment are necessary to obtain a good representation of the IPv6 space – ideally on a daily basis. The big difference between IPv4 and IPv6 is also illustrated when we compare both in Figure 5b, which shows the average intra-month catchment stability over all months analyzed.

4.2 Diving into catchment changes

In the previous section, we investigated the churn of prefixes in terms of percentages and observed that after 2 months, the original catchment was no longer representative of the current state of the deployment. The obvious follow-up question is: *between which nodes do prefixes shift the most?*

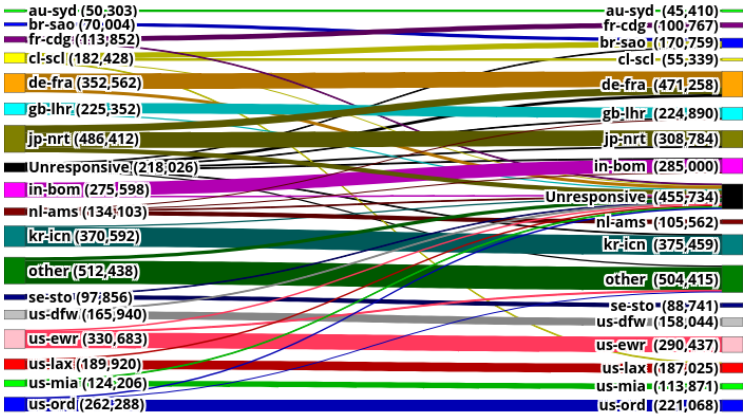


Fig. 6: Catchment shifts between April 1 (left) and July 1 (right), filtering on changes of $>$ 10,000 prefixes for the 16 largest sites.

We compared the catchment of April 1st with that of July 1st and analyzed the shift of client prefixes among different catchments over this time interval. Figure 6 shows the catchment changes for the 16 largest Vultr sites in terms of catchment, with the remaining 16 grouped into 'other'. Only shifts with more than 10,000 prefix changes are shown in the graph. The left side of the graph represents the catchment on April 1st, while the right side represents July 1st.

The majority (72%) of prefixes did not change sites over the analysis period. However, a significant portion (28%) did switch, with many of these (18%) moving between different continents over large geographic distances.

For some sites, we noticed a substantial number of catchment changes (e.g., br-sao, cl-scl, jp-nrt, de-fra, us-ewr), indicating possible major re-routing events leading to significant flows of clients shifting between catchments.

To assess whether such catchment shifts are caused by instances of geographical transfer of IP addresses, we compared IP2Location data [7] for April 1st and July 1st that shows a negligible number (0.8%) of prefixes moved location.

A deeper analysis of these large flows shows that they originate from a few ASes, suggesting that while catchment changes are considerable and lead to inter-continental routing, they are confined to a few ASes. This implies that the situation could be easily addressed by the anycast operator using BGP path manipulation (*e.g.,* path prepending). We leave this to future work.

We argue that this risk of inter-continental routing is a fundamental reason for frequent active catchment analysis for anycast services. This would help network operators avoid severe performance degradation due to traffic being rerouted halfway across the globe instead of being directed to the nearest topological and geographical PoP.

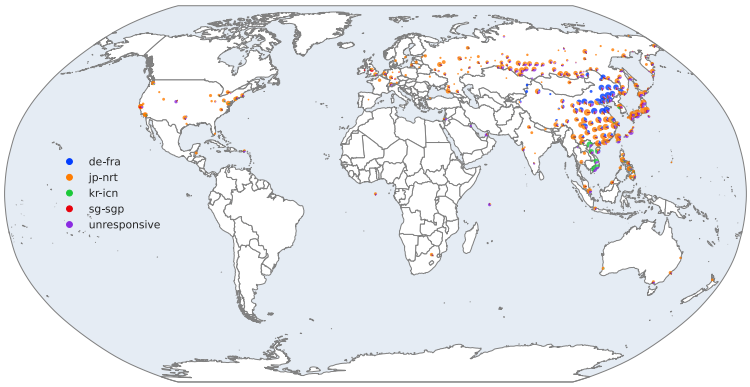


Fig. 7: Visualization of the geolocation of prefixes in the catchment of jp-nrt on April 1st, and where they re-routed to on July 1st indicated with pie charts.

4.3 Zooming in further: a case study of jp-nrt

Among our major sites, in terms of catchment, we observe the largest change in catchment for jp-nrt (our anycast site in Tokyo, Japan). For this site we observe a catchment of 486 k /24 IPv4 prefixes on April 1st, and a catchment of 308 k prefixes on July 1st (a change of -36%). The change in catchment can be seen in Figure 6, where we see most traffic went either to de-fra or became unresponsive.

Figure 7 zooms in on this case. We display the locations of the addresses that route to jp-nrt on April 1st, using *ip2location*. The map shows that the majority of prefixes originate from east Asia, as expected. Furthermore, we observe many prefixes in Russia route to Tokyo. Additionally, we find prefixes located in distant regions (e.g., USA, and Western Europe) routing to Tokyo. However, for these distant prefixes we measure low latency responses that would violate the speed-of-light, indicating *ip2location* likely reports incorrect locations for these instances. Next, using the colors of the circles we show where these prefixes routed to on July 1st (after the 36% reduction in catchment), the mapping of color to anycast site is displayed in the legend. We observe that most changes in catchment are regional, e.g., prefixes in Thailand change to Seoul, prefixes in north eastern China switch to Frankfurt. For those that remained in Tokyo's catchment we observe many are located in Russia and Eastern Asia. Investigating prefixes that switched catchment, we observe the large changes of Chinese prefixes originate from China Unicom (AS4837, AS4808).

Finally, we show the difference in RTT in Table 1 grouped by catchment change. We find that the Chinese prefixes that switched to de-fra more than tripled in RTT, whilst those that remained in Tokyo's catchment had a slight reduction in mean RTT. Furthermore, prefixes that switched to kr-icn see a slight increase in mean RTT, whereas those that switched to sg-sgp had a substantial

Table 1: Differences in RTT for the largest catchment changes observed at jp-nrt between April 1st and July 1st.

New site in July	Number of prefixes	Mean RTT to JP-NRT	Mean RTT to new site
de-fra	125,418	73.0 ms	255.3 ms (+250%)
jp-nrt*	296,304	89.5 ms	83.4 ms (-7%)
kr-icn	9,291	92.7 ms	99.4 ms (+7%)
sg-sgp	1,134	59.5 ms	43.5 ms (-27%)
Total	432,147	84.0 ms	141.1 ms (+68%)
unresponsive	78,153	81.7 ms	*n/a*

decrease in RTT. Evaluating the *total* change in catchment for these groups, we find the overall mean RTT to nearly double.

Following July 1st, our commercial upstream provider withdrew our announcement at Tokyo, without notice, leaving the site inactive for several weeks. For this reason, we did not evaluate behavior of this catchment after this date.

This case study shows catchment changes that can largely be grouped into regional changes caused by few ASes. Furthermore, we find catchment changes can both improve and worsen mean latency, leading in some cases to ruinous cross-continental changes in catchment. This case study underscores the need to regularly map catchments to detect such changes promptly allowing operators to take steps to implement traffic engineering measures to address latency inflation (a problem also discussed by, *e.g.,* Calder et al. [2]).

5 Conclusions

In this work, we put long-term catchment stability on an empirical footing to help network operators establish a baseline for their catchment scanning intervals. Our analysis showed that catchments are stable for both IPv4 and IPv6 in the short term (day-to-day operations). In the long term, IPv4 catchments are more stable than their IPv6 counterparts, with stability lasting over 3 weeks, assuming an error rate of 10%. In contrast, the representativeness of the IPv6 catchment decreases after just 2 days, mostly due to a lack of stable hitlists for IPv6. To maintain a good representation of the status of their anycast catchment, we recommend that network operators repeat catchment measurements on a weekly basis for IPv4 and a daily basis for IPv6. These measurements will help them to promptly identify and mitigate cases of suboptimal routing. Next steps are understanding what causes shifts in catchment by *e.g.,* looking at BGP route changes and path changes using traceroute. Furthermore, zooming out, understanding catchment stability is a key ingredient to studying the use of catchment data to detect spoofing and off-site traffic detection, as first suggested by De Vries *et al.* [17]. We plan to use the lessons we learned about catchment stability to study spoofing and off-site traffic detection in depth in future work.

References

1. Vultr data center worldwide locations. https://www.vultr.com/features/datacenter-locations/ (2024)
2. Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., Padhye, J.: Analyzing the Performance of an Anycast CDN. In: Proceedings of the 2015 Internet Measurement Conference. pp. 531–537. IMC '15, Association for Computing Machinery, New York, NY, USA (2015)
3. Cicalese, D., Augé, J., Joumblatt, D., Friedman, T., Rossi, D.: Characterizing ipv4 anycast adoption and deployment. In: Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies. CoNEXT '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2716281.2836101, https://doi.org/10.1145/2716281.2836101
4. Fan, X., Heidemann, J., Govindan, R.: Evaluating Anycast in the Domain Name System. In: 2013 Proceedings IEEE INFOCOM. pp. 1681–1689 (2013). https://doi.org/10.1109/INFCOM.2013.6566965
5. Gasser, O., Scheitle, Q., Foremski, P., Lone, Q., Korczyński, M., Strowes, S.D., Hendriks, L., Carle, G.: Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In: Proceedings of the Internet Measurement Conference 2018. pp. 364–378. IMC '18, Association for Computing Machinery, New York, NY, USA (2018)
6. Giordano, D., Cicalese, D., Finamore, A., Mellia, M., Munafò, M.M., Joumblatt, D.Z., Rossi, D.: A First Characterization of Anycast Traffic from Passive Traces. In: Botta, A., Sadre, R., Bustamante, F.E. (eds.) Traffic Monitoring and Analysis - 8th International Workshop, TMA 2016, Louvain la Neuve, Belgium, April 7-8, 2016. IFIP (2016), http://dl.ifip.org/db/conf/tma/tma2016/tma2016-final30.pdf
7. IP2Location: IP Address to IP Location and Proxy Information. https://www.ip2location.com/, [Accessed 12-07-2024]
8. Koch, T., Katz-Bassett, E., Heidemann, J., Calder, M., Ardi, C., Li, K.: Anycast in Context: a Tale of Two Systems. In: Proceedings of the 2021 ACM SIGCOMM 2021 Conference. pp. 398–417. SIGCOMM '21, Association for Computing Machinery, New York, NY, USA (2021)
9. Moura, G.C., Schmidt, R.d.O., Heidemann, J., de Vries, W.B., Müller, M., Wei, L., Hesselman, C.: Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. In: Proceedings of the 2016 Internet Measurement Conference. pp. 255–270. IMC '16, Association for Computing Machinery, New York, NY, USA (2016)
10. Partridge, C., Mendez, T., Milliken, W.: Host Anycasting Service. RFC 1546, IETF (Nov 1993), http://tools.ietf.org/rfc/rfc1546.txt
11. Quan, L., Heidemann, J., Pradkin, Y.: Trinocular: Understanding Internet Reliability through Adaptive Probing. In: Proceedings of the ACM SIGCOMM 2013 Conference. pp. 255–266. SIGCOMM '13, Association for Computing Machinery, New York, NY, USA (2013)
12. van Rijswijk-Deij, R., Jonker, M., Sperotto, A., Pras, A.: A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements. IEEE Journal on Selected Areas in Communications **34**(6), 1877–1888 (2016). https://doi.org/10.1109/JSAC.2016.2558918
13. RSSAC: Root Server Technical Operations Association. https://root-servers.org (2024)
14. Schomp, K., Bhardwaj, O., Kurdoglu, E., Muhaimen, M., Sitaraman, R.K.: Akamai DNS: Providing Authoritative Answers to the World's Queries. pp. 465–478. SIGCOMM '20, Association for Computing Machinery, New York, NY, USA (2020)

15. Sommese, R., Bertholdo, L., Akiwate, G., Jonker, M., van Rijswijk-Deij, R., Dainotti, A., Claffy, K., Sperotto, A.: MAnycast2: Using Anycast to Measure Anycast. In: Proceedings of the ACM Internet Measurement Conference. pp. 456–463. IMC '20, Association for Computing Machinery, New York, NY, USA (2020)
16. USC/LANDER project: Internet Addresses Survey Dataset. `http://www.isi.edu/ant/lander` (2024), PREDICT ID: USC-LANDER/internet_address_hitlist_it108w-20240711.
17. de Vries, W.B., Aljammāz, S., Rijswijk-Deij, R.v.: Global-Scale Anycast Network Management with Verfploeter. In: NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. pp. 1–9 (2020)
18. de Vries, W.B., de O. Schmidt, R., Hardaker, W., Heidemann, J., de Boer, P.T., Pras, A.: Broad and Load-aware Anycast Mapping with Verfploeter. In: Proceedings of the 2017 Internet Measurement Conference. pp. 477–488. IMC '17, Association for Computing Machinery, New York, NY, USA (2017)