Characterizing Anycast Flipping: Prevalence and Impact

Xiao Zhang ° *, Shihan Lin*, Tingshan Huang†,
Bruce M. Maggs*, Kyle Schomp°, Xiaowei Yang*

Duke University*, Cisco ThousandEyes°, Akamai Technologies†

Abstract. A 2016 study by Wei and Heidemann showed that anycast routing of DNS queries to root name servers is fairly stable, with only 1% of RIPE Atlas vantage points "flipping" back and forth between different root name server sites. Continuing this study longitudinally, however, we observe that among the vantage points that collected data continuously from 2016 to 2024 the fraction that experience flipping has increased from 0.8% to 3.2%. Given this apparent increase, it is natural to ask how much anycast flipping impacts the performance of everyday tasks such as web browsing.

To measure this impact, we established a mock web page incorporating many embedded objects on an anycast-based CDN and downloaded the page from geographically distributed BrightData vantage points. We observed that packets within individual TCP flows almost always reach the same site, but different flows may flip to different sites. We found that 1,988 (10.9%) of 18,294 <vantage point, anycast IP> pairs suffer from frequent anycast flipping (i.e., a vantage point is directed to sites other than the most visited one of the anycast IP more than 50% of the time) and that 1,030 of these (5.6% of the total) suffer a median increase in round-trip time larger than 50 ms when directed to a site other than the most visited one.

We then used Mahimahi to emulate downloads of popular web sites, randomly applying the above-mentioned flipping probability (50%) and flipping latency penalty (50 ms) to CDN downloads. We found, for example, that there was a median increase in the First Contentful Paint metric ranging, across 3 vantage points and 20 web sites, from 20.7% to 52.6% for HTTP/1.1 browsers and from 18.3% to 46.6% for HTTP/2 browsers. These results suggest that for a small, but not negligible portion of clients, the impact of anycast flipping on web performance may be significant.

1 Introduction

IP anycast [35,31] is a routing paradigm that splits traffic among a set of physical sites. Operators advertise the same IP prefix – via the Border Gateway Protocol (BGP) [19] – from each site and traffic from clients to the prefix is routed to any one of the sites, distributing the traffic load across the sites. Further, IP anycast presents the opportunity to improve performance by routing clients to a proximal

* Xiao Zhang was with Duke University at the time this work was conducted. He is now with Cisco ThousandEyes.

site [14,17,51]. For these reasons, IP anycast is widely adopted by large-scale network services, such as the Domain Name System (DNS) [30,43], content delivery networks (CDNs) [48] and DDoS mitigation services [32,45]. Thus, the efficient operation and performance of anycast networks is critical to today's Internet.

Clients of anycast services are commonly organized according to which sites they reach, and the set of clients reaching the same site is known as that site's "catchment". It is possible for a client to switch catchments. In the past, operators have raised concerns that such anycast "instability" could interrupt connection flows [28]. Because connection-oriented protocols typically store state at both ends of the connection, a change in site during a flow would break that flow. The existence of major services relying on anycast, however, belies the prevalence of broken flows. Indeed, previous work [49] demonstrates that changes in site occur rarely in measurements from RIPE Atlas [44] probes to the DNS root nameservers.

The same study, however, observes that 1% of the vantage points measured did "flip" frequently between catchments, sometimes changing with *each* measurement. In follow-up work [50], the authors show that flipping within TCP flows is rarer than flipping overall (impacting 0.15% of vantage points), and provide a plausible explanation that flipping occurs per-flow, rather than per-packet, so that all packets within a single flow reach the same site, preserving the connection. So, while flipping does occur, it is not observed to interrupt flows.

In our own measurements of anycast services, we observed high variability of round-trip-time (RTT). Intrigued, we set out to discover the source of the variability and found ourselves retreading the topic of flipping. While flipping is rare, using the same methodology as in [49] we found that for RIPE Atlas probes querying root name servers, flipping has more than quadrupled from 0.8% in 2016 to 3.2% in 2024, among the vantage points that collected data continuously over that time. In these experiments, a probe issues a query every four minutes, and a flip is said to occur if two consecutive queries by the same probe are routed to different root name server sites. Following the methodology in [49], a probe is said to be "flipping" if it averages at least one flip every ten minutes.

Investigating the cause of the increase through a longitudinal study, we find that increasing numbers of anycast sites – adding to the possibilities for flipping – contribute to the prevalence of flipping in 2024. With these changing network realities, we argue that anycast flipping is a more significant issue now, warranting further study of its impact on web performance.

Next, by downloading a mock web page with many embedded objects from a major anycast CDN through the BrightData residential proxy service [3], we confirm the finding that flipping rarely breaks connections and appears to occur per-flow. Furthermore, we find that the incidence of flipping from the BrightData vantage points to the CDN sites is perhaps even higher than from the RIPE Atlas probes to the root name servers. Because the CDN experiment differs from the DNS experiment (e.g., we didn't download the page every four minutes), we cannot adopt the same definition of flipping for both experiments. Instead we say that a vantage point is flipping if more than 10% of TCP connections are sent to a site other than the most common CDN site for

that vantage point. With this definition, in the CDN experiment we found that 6,028 of 18,294 (33.0%) <BrightData vantage point, CDN anycast IP> pairs were flipping.

The CDN experiments also revealed that the round-trip time from a vantage point to the sites its requests are sent to can vary widely. Hence, we set out to measure the impact of flipping and the corresponding variations in latency on the time to download and render popular web sites.

In summary, this paper makes the following contributions:

1. We show that when querying DNS root name servers from RIPE Atlas probes, the prevalence of anycast flipping has increased from only 0.8% of probes in 2016 to 3.2% in 2024. We confirm the previous finding that flipping rarely breaks connections, and that flipping commonly occurs at the granularity of an entire flow. Further, using the modified 10% criteria for HTTP experiments, we find that flipping is also common when issuing requests from the BrightData residential proxy network [3] to a major anycast CDN, with 32.9% of <proxy, CDN anycast IP> pairs flipping.
2. We show that flipping can significantly deteriorate the RTT of a client to the website due to connecting to a further CDN site. For example, 20% of these flipping vantage points have a median increase in RTT larger than 101.3 ms, and 10% have a median RTT increase larger than 167.7 ms.
3. We demonstrate that for clients that experience very frequent flipping and large latency penalties, the impact on web download performance can be significant. For example, we performed trace-driven emulations of 20 popular web sites with Mahimahi [33], extending it so that we could randomly apply a flipping probability of 50% to entire TCP flows and a flipping latency penalty of 50 ms. In this experiment, there was a median increase of more than 20.7% - 52.6% in the First Contentful Paint metric for HTTP/1.1 browsers and 18.3% - 46.6% for HTTP/2 browsers.

The current prevalence of anycast flipping, which is already significant and appears to be increasing, combined with the high latency penalties experienced by some clients when flipping occurs, calls for careful optimization of anycast deployments and perhaps even improved routing protocol design.

Ethical considerations Active measurements such as issuing DNS queries and downloading web pages cause load on the Internet infrastructure. As discussed later in the paper, we mitigated the impact of our experiments by issuing measurements at low rates or a small number of times. Our analysis of the frequency of DNS-query flipping is based on publicly available RIPE Atlas data, and did not require us to initiate any measurements. Our measurements of the anycast CDN were directed at a mock web site that was hosted using a free service provided by the CDN, the residential proxy is supported by Bright Initiative [7]. This work raises no other ethical issues.

Data and code Upon publication, the authors will make all data collected in this research and all code used to collect or process that data publicly available. The authors will also provide such data and code to the program committee upon request.

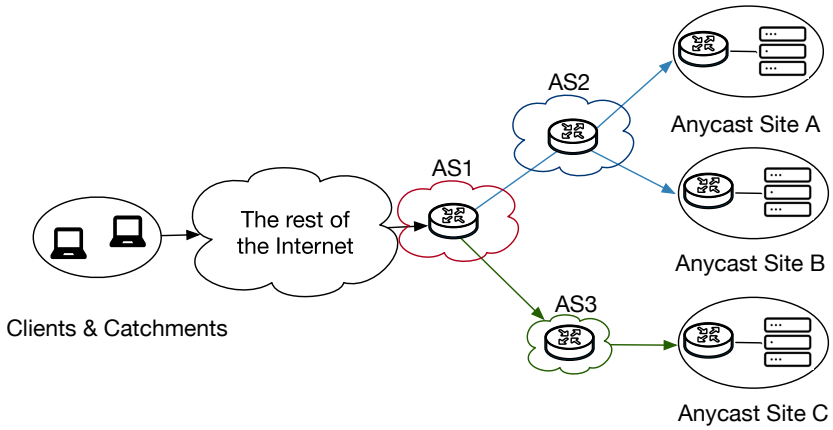


Fig. 1: **An IP anycast deployment.**

2 Background

In this section, we use network traces to illustrate several examples of anycast flipping in the Internet.

Figure 1 shows a simple anycast network with three sites, each one serviced by a single upstream autonomous system (AS). Routers at each of the sites peer with routers in the upstream AS and advertise a route to the same IP prefix. Upon receiving the advertisements, the upstream ASes propagate them further to other ASes. If a router receives multiple routes to the IP prefix (e.g., the router in AS1), it applies BGP best path selection [19] to choose a route to reach that prefix. Best path selection is a cascading comparison between metrics of the available routes, including – among others – the AS path length. Traffic from clients of the service to the IP prefix is then routed to different sites via the emergent properties of the different path selection choices made by routers in the Internet. The set of clients all reaching the same site is the *catchment* of the site.

Over time, catchments can change for a variety of reasons. One of the most straightforward is route updates. Upon each route advertisement/withdrawal, routers recompute the best path, which may be different from the previous one and originate at a different site. Wei and Heidemann [49] argue that updates are unlikely to cause frequent catchment changes due to BGP route flap damping (RFD) [47], although a sophisticated analysis by Gray et al. [22] provides a lower bound of only 9.1 on the percentage of ASes that implement RFD. But flipping can have other causes as well, including IGP Equal-Cost Multipath (ECMP), and BGP Multipath.

In this section we illustrate two examples that were collected from RIPE Atlas probes that we observed flipping. More details on our RIPE Atlas datasets are provided in Section 3.

In the first example, shown in Figure 2, flipping takes place at the border of two ASes. The figure shows multiple traceroutes from RIPE Atlas probe #1002101 to the DNS C-root on Jan-02-2023. The traceroutes have been merged so that hops appearing in multiple traceroutes are represented by a single node in the graph. The

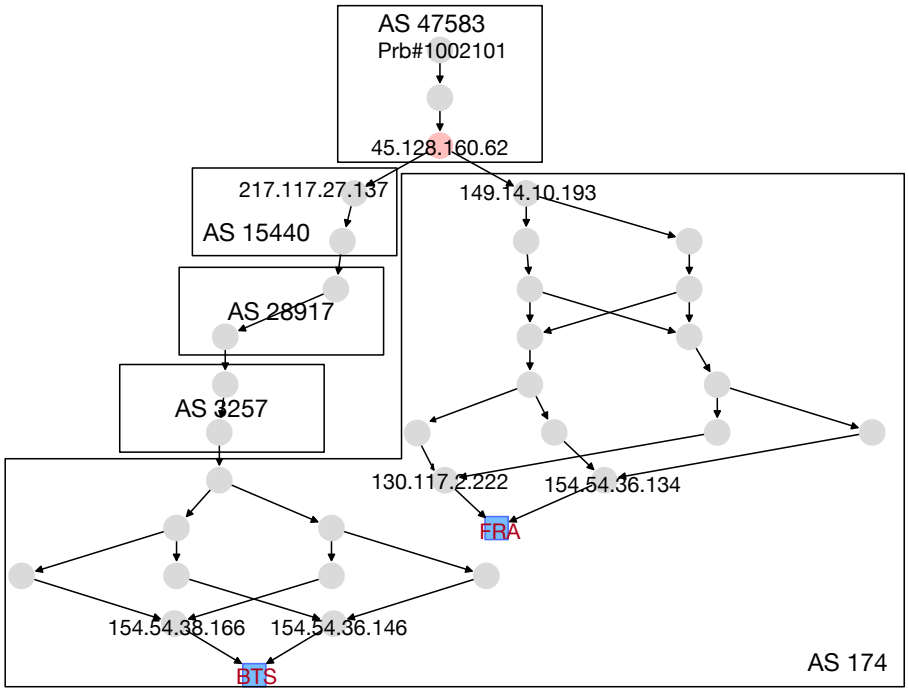


Fig. 2: Flipping at the border between two ASes

probe flips between anycast sites in Frankfurt (FRA) and Bratislava (BTS). While the traceroutes produce numerous paths to the target anycast IP address, the third hop – the last in AS47583 – visually splits the paths among the two sites. Interestingly, the hops immediately after AS47583 are from two different ASes.

We are not sure what mechanism is responsible for flipping in this example. Originally, BGP Multipath required (among other things) the AS path of an alternate route to be strictly identical to the primary route in both the length and the sequence of AS numbers. As [26] notes, however, router configuration settings are often provided to relax the sequence requirement, e.g., Cisco's `multipath-relax` setting [1] and Juniper's `multiple-as` setting [2]. But in this example both the length and the sequence are different.

Figure 3 shows an example in which the flipping is internal to an AS. Probe #2121 flips between anycast sites of C-root in Frankfurt (FRA), and Paris (PAR) on Jan-02-2023. AS174 advertises routes to AS2914 in at least 3 places. Hop 81.25.197.1 forwards flows further within AS2914 that ultimately traverse different egress points to reach the different anycast sites. We speculate that AS174 applies ECMP internally to load balance traffic within the network, contributing to the flipping we observe.

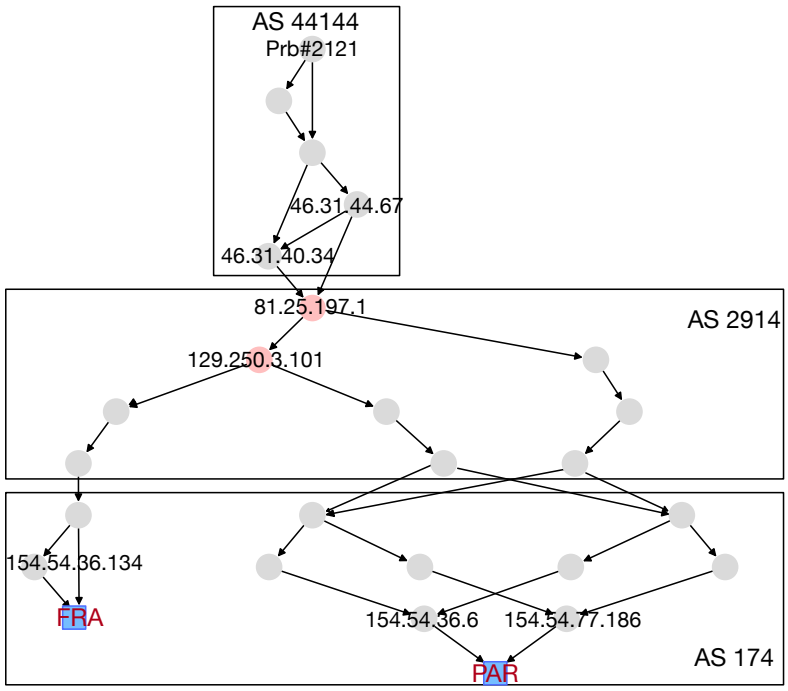


Fig. 3: Flipping internal to an AS

3 Prevalence & Impact: Root DNS

In this section, we use RIPE Atlas's built-in measurements [39] to root DNS servers to characterize the prevalence of anycast flipping and its impact on the Atlas probe's RTT to each root DNS server. We first describe the measurement infrastructure and datasets we use. Then, we present how we infer anycast flipping from the datasets. Finally, we present the trend and degree of anycast flipping from RIPE Atlas probes to root DNS servers between 2016 and 2024.

3.1 RIPE Atlas's DNS CHAOS Queries

RIPE Atlas [44] is a global measurement infrastructure that has more than 11,000 active probes world-wide. It conducts several periodic measurements to all 13 root DNS servers and archives the measurement results for researchers to study the properties of the Internet. Specifically, we use the DNS CHAOS [16] measurement for this study. In this measurement, all probes send a DNS CHAOS query for the TXT field of the domain name `hostname.bind` to each root DNS server every four minutes. The probe records the response that it receives. A response to a CHAOS query from a root server includes a site identifier (ID), which we use to identify the catchment site of the query.

Datasets To evaluate the trend of anycast flipping, we use the DNS CHAOS responses collected on the first Monday (24-hours) of every half-year from the last

half year of 2016 to the last half year of 2024, a total of 221 datasets. For comparison to [49], we similarly use all probes that RIPE reports active on the day of the dataset. For the first dataset collected on Jul, 04, 2016, there are 8,958 active probes; and for the most recent dataset collected on Jul, 01, 2024, the number of active probes increases to 12,724. The variability in active probes over the course of our longitudinal study could skew our findings on prevalence of anycast flipping, so separately we also study the 555 to 871 probes that are active across all datasets, per root DNS.

3.2 Root DNS Site Discovery

We parse the site ID out of the TXT record included in a server's response to determine the catchment site of a probe's CHAOS query and we use the change of the catchment site to detect anycast flipping (Section 3.4).

Each root DNS server uses a customized naming scheme to embed a site ID in the TXT record. For example, a site ID in a response from A-root can be 'nnn1-nlams-1a' or 'rootns-fra5', each consisting of several sections. The first section (e.g., 'nnn1' or 'rootns') indicates which DNS software is running on the responding server. The second section encodes the location of the site. In the first example, 'nlams' is the UN/LOCODE [20] for Amsterdam, Netherlands, while in the second example, 'fra' is the IATA [23] airport code for Frankfurt, Germany. The third section in the first example indicates the specific server/replica ID at the Amsterdam site. Similarly, the number '5' in the second section of the second example specifies the ID of the server located at the 'fra' site.

We manually examine the CHAOS responses from each root server and construct regular expressions to extract the site ID field from those responses. In addition, the site naming scheme of each root has also changed over the years. For example, L-root's scheme changed three times in the last eight years. Therefore, we use different regular expressions per root and per time period to extract the site ID from the TXT field of each CHAOS response. We note that some CHAOS replies do not follow any naming convention and appear to come from record injection attacks [38]. An example is a TXT record that includes the string `byaazbknliphsiiy.vla.yp-c.yandex.net`. The regular expressions we construct automatically filter such responses. For the datasets we use, we filter 2.0%−2.2% of such responses. We include the regular expressions we construct in Appendix A.1. While there may be carefully crafted injections that also match the regular expressions, they would likely only impact our analysis to a very limited extent.

3.3 Geo-Locate a Catchment Site

In addition to identifying a catchment site, we also aim to geo-locate where the site is at the city level. This geo-location information enables us to measure the distance between catchment sites that a probe flips between. Since most root servers include geo hints such as UN/LOCODE or IATA codes in their site IDs, we first extract those geo-codes from the site IDs we obtain in the previous step. Second, we use a geo-location method to confirm that 1) the geo-code included in a site's identifier is accurate, and 2) resolve the locations of the sites that either contain erroneous geo-codes or do not contain valid geo hints. The second step is important because a site

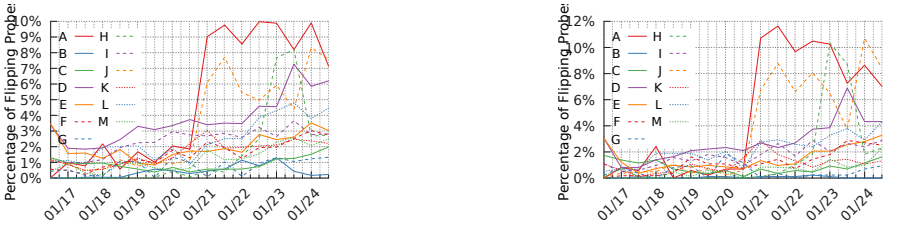
ID may include a wrong code. For example, a site used by A-root has 'tko' in its site identifier, which is the IATA code for the Tlokoeng airport in South Africa. However, using the geo-location method, we find that the site is actually located in Tokyo.

Our geo-location method works as follows. First, we pick the top three probes that have the lowest RTTs extracted from the same DNS request to a site. Because the RTT between two points at a distance of 100km trip is approximately 1 ms at the speed of light in fiber, if the RTT of each probe to the site is less than 5 ms, and the geo-code in the site's identifier specifies a location that is within a radius of 500 km of one of the probes, we consider the site's geo-code accurate and use the geo-code as the site's location. For the remaining site IDs, we check whether the site identifiers contain any geo-hint that matches one of the three closest probes' locations. For example, a site of A-root contains the string 'elpek' in its ID. Although it complies with the format of a UN/LOCODE, it is not a valid UN/LOCODE. But the sub-string, 'pek' is the IATA airport code for Beijing, China, and the three probes within 5 ms of this site confirm that the site is in Beijing. Finally, for the remaining unresolved sites, we use IP-geo information [24] of the penultimate hop in a traceroute measurement from each of the three-closest probes that only targeting to this site to locate the site. For example, a site ID 's1.org' from I-root contains no valid geo-hint and is resolved in this final step. We verified that all of the site geo-locations developed using this method match the reported locations of root server sites from [6].

We use the dataset obtained on Jan 02, 2023 as an example to show how many sites we are able to resolve at each step. We extract a total of 1,014 site IDs. Among them, we confirm 808 of them as valid geo-codes in the first step. We are able to resolve 198 sites' locations in the second step. The final step (traceroute) resolves the remaining eight sites. Other datasets have similar results.

3.4 Estimating Anycast Flipping

After we obtain the site ID and the geo-location of each catchment site a probe reaches, we are able to quantify anycast flipping. We use the same anycast flipping detection and counting mechanism as in [49]. For each probe and for each dataset, we construct a time-series vector that consists of the catchment sites a probe reaches. If the catchment sites in two consecutive CHAOS responses are different, we count it as a flip. We then calculate the mean time between two flips across the whole day for each probe. Following the methodology in [49], if the mean flipping time is less than 10 minutes (1 flip per 2.5 rounds), then we consider this probe to experience anycast flipping. While we could devise other approaches to measuring anycast flipping, our goal is to revisit the findings in [49] and extend them to determine the relative prevalence of anycast flipping since 2016. To verify that we've reproduced the mechanism accurately, we compare the flipping results we obtain using the same dataset as used in [49] (Aug-01-2016) and confirm the results are almost identical. For example, on that day, C-root had a flipping fraction of 1.2% in our measurement, and it is the same value in [49], similar findings for the other roots. Therefore, we conclude that we are measuring the same phenomenon as in [49].



(a) *All RIPE Atlas probes*

(b) *RIPE Atlas probes that are active throughout the past eight years*

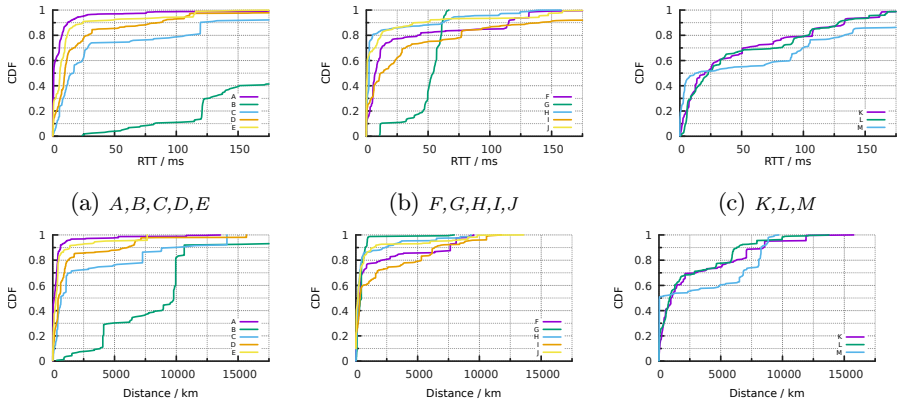
Fig. 4: *The percentage of RIPE Atlas probes that experience anycast flipping to root DNS servers over the past eight years. Left: All probes whose measurements are available. Right: the common subset of the probes that are active throughout the entire eight years.*

3.5 Longitudinal View of Anycast Flipping

Now, we describe our reproduction of the results from the 2016 study [49] and our longitudinal study of anycast flipping since then. Among all active probes, our longitudinal study finds that the percentage of RIPE Atlas probes that experience anycast flipping has increased from about 1.1% to 3.5% within the past eight years, when averaged across the DNS roots. Since the RIPE Atlas probe deployment has changed over those years, we also measure the increase in anycast flipping by holding the probes constant to only the subset of probes that are active for the entirety of the study, since this subset should be more likely to stay stable in the same network and geographical area than the other. This number varies per DNS root, between 555 for F-root and 871 for H-root. Again when averaged across the DNS roots, we find that anycast flipping has increased from 0.8% to 3.2% among the probes active throughout the entire study.

We present the longitudinal percentage of anycast flipping probes in Figure 4a and Figure 4b, for all RIPE Atlas probes and the continuously active probes, respectively. As the trend is similar for both, we focus on describing the findings for all probes. For nine of the 13 roots, the probes have a significant increase in anycast flipping during our measurement period. The most noticeable changes occur with A-root and J-root. Eight years ago, anycast flipping with both roots was minimal, but now, A-root incurs one of the largest amounts of anycast flipping among RIPE Atlas probes, from 0.0% (only very few probes flipped at that time) to 7.1%. J-root sees a similar increase, although not as pronounced. Interestingly, both had a significant step between the second half year of 2020 and the first half year of 2021. We investigate A-root to better understand the causes.

A-root anycast flipping drastically increased from 2.0% to 9.8% between 2020 and 2021. On Jul-06-2020, RIPE Atlas probes received CHAOS TXT records that mapped to 21 sites, but half a year later the records mapped to two additional sites, 'mnz' and 'wie'. Further, 88.9% of the probes that started flipping in 2021 but were not in 2020 reached at least one of the two new sites. We suspect that A-root deployed the two new sites at this time and that the introduction of the new sites likely created new opportunities for BGP Multipath (see Section 2).



(a) *A,B,C,D,E* (b) *F,G,H,I,J* (c) *K,L,M*

(d) *A,B,C,D,E* (e) *F,G,H,I,J* (f) *K,L,M*

Fig. 5: *Extra RTT and Extra Distance Caused by Anycast Flapping for Anycast Flipping Probes*

3.6 Does Flipping Break TCP Connections?

The prevalence of anycast CDNs suggests that anycast flipping does not tear down TCP connections, and the anycast flipping study [49] in 2016 found that very few RIPE Atlas probes consistently failed to build a TCP session with the J root name server and a small number of those probes also suffer from UDP anycast flipping.

We use the RIPE Atlas built-in measurement that sends TCP DNS queries to obtain the SOA field of route DNS servers to further confirm this phenomenon. Using a dataset obtained on Jan-02-2023, we find that 97.4% - 98.0% of the probes can successfully establish TCP connections with a root DNS server. Only 220 probes cannot establish TCP connections to any root DNS server and among those probes, at most 8 of them experience anycast flipping to some root DNS servers. The very small overlap between probes that cannot establish a TCP connection and probes that experience anycast flipping suggests that anycast flipping is unlikely to break TCP connections for the majority of Internet paths.

3.7 Impact on RTTs

For each of the RIPE Atlas probes (154 - 1243 for different roots) that we observed flipping on Jan-02-2023 to a root DNS server, we compute the median RTT towards each of the anycast sites that they flip between. We then take the difference in the RTT to determine how much extra RTT flipping adds. Note that probes can flip between more than 2 sites – for A-root, 205 probes do. In that case, we take the difference between the minimum and maximum median RTTs to the sites, to determine the worst case extra RTT that flipping adds. Figure 5 shows the distribution per root of the extra RTT. In addition, we show the percentage of the probes whose difference in RTT exceeds 50ms and that flip among more than 2 sites in Table 1.

Root	Probes that flip	>50ms difference	>2 sites
A	1243	38 (3.06%)	205 (16.49%)
B	162	156 (96.30%)	2 (1.23%)
C	154	40 (25.97%)	0 (0.00%)
D	573	84 (14.66%)	121 (21.12%)
E	309	24 (7.77%)	40 (12.94%)
F	249	45 (18.07%)	22 (8.84%)
G	164	106 (64.63%)	0 (0.00%)
H	965	114 (11.81%)	0 (0.00%)
I	334	83 (24.85%)	16 (4.79%)
J	746	71 (9.52%)	76 (10.19%)
K	265	92 (34.72%)	30 (11.32%)
L	542	177 (32.66%)	92 (16.97%)
M	288	130 (45.14%)	0 (0.00%)

Table 1: Anycast flipping impact *(>**50ms difference**: Flipping that causes more than 50 ms round-trip-time difference between sites. >**2 sites**: Flipping that reaches more than 2 different anycast sites)* .

For B-root and G-root, the difference in RTT is extreme: nearly 80% of the flipping RIPE Atlas probes experience extra RTT of over 120ms/50ms to B-root/G-root, respectively. More probes flip reaching A-root than any other root, but the extra RTT in A-root is minimal: the extra RTT for 90% (1118 out of 1243 probes) of the probes less than 12ms, indicating that flipping in A-root occurs between nearby sites. For the other roots, the majority of probes experience modest extra RTT. Overall, though, 24 to 177 probes (0.2% - 1.4% of all probes) experience more than $\geq$50ms extra RTT to at least one root.

We measure the distance between a probe and its catchment site. In Figure 5, we show the distribution of the extra distance corresponding to a probe's flipping sites with which we compute the extra RTT. As we can see, the long extra RTTs often correspond to long geographic distance between a probe's catchment sites. In G-root, however, the large extra RTT does not correspond to similarly large extra distance. We're uncertain why anycast flipping adds so much RTT in this case. In one extreme example, we highlight the case of RIPE Atlas probe #6506 which is reportedly located in Singapore. This probe flips between a C-root site located also in Singapore and another site in Los Angeles, USA.

4 Prevalence & Impact: Anycast CDN

We have observed anycast flipping in the root DNS IP anycast system. Does anycast flipping also impact anycast CDNs? To answer this question, we study anycast flipping in a major anycast CDN. In this section, we use BrightData [3], a proxy service provider, to explore the prevalence of anycast flipping using different vantage points than RIPE Atlas probes and with a new target: an anycast CDN.

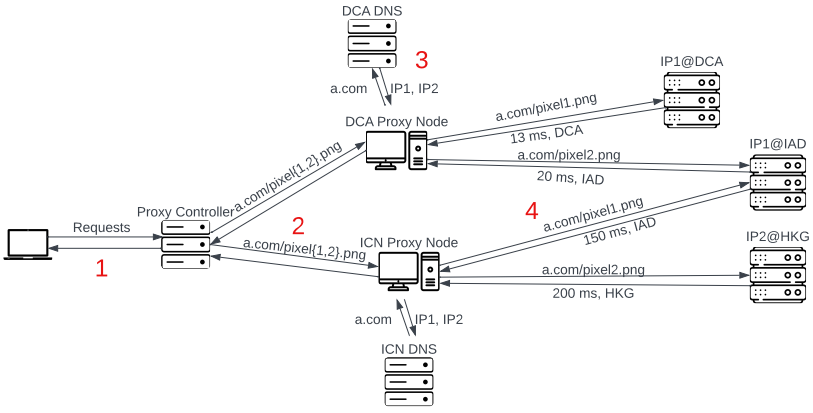


Fig. 6: This figure shows how we measure anycast flipping to an anycast CDN. The measurement machine connects to the proxy controller and selects the proxy nodes that send an HTTP request for the mock webpage we create. When a proxy node contacts its DNS server, it receives two IP addresses returned by the anycast CDN's DNS. A proxy node chooses either one or two IPs to send the HTTP requests. When anycast flipping happens, the DCA (Arlington, VA) proxy node reaches two CDN edge servers (DCA-Arlington, VA server and IAD-Dulles, VA server) when using the IP address IP1.

4.1 Infrastructure

The RIPE-Atlas-based measurements can show us anycast flipping to the root name servers. To our knowledge, however, no existing measurements cover anycast flipping to CDN servers. Therefore, we utilize a residential proxy service provider, BrightData [3], to measure anycast flipping to one of the largest anycast CDNs. This residential proxy service has previously been used in other measurement studies, e.g. [15]. According to their website description, BrightData's infrastructure consists of more than 72 million proxy nodes. In our experiments, we uncover 9,568 unique vantage points (distinguished by IP address) that span 143 different countries and 1,912 different ASes.

4.2 Methodology

To measure anycast flipping, we build a mock webpage, consisting of one main page embedded with 50 one-pixel images. Next, we configure this webpage and all embedded images to be served by the anycast CDN and make them cacheable by the CDN. Therefore, a web client can retrieve them directly from the CDN's edge servers.

Figure 6 shows our experimental setup. Our measurement machine connects to BrightData's proxy controller and instructs the controller to select geographically distributed proxy nodes for experimentation. To simulate the webpage retrieval process from the anycast CDN server, we instrument `curl` to fetch the 50 pixel images through the proxy node.

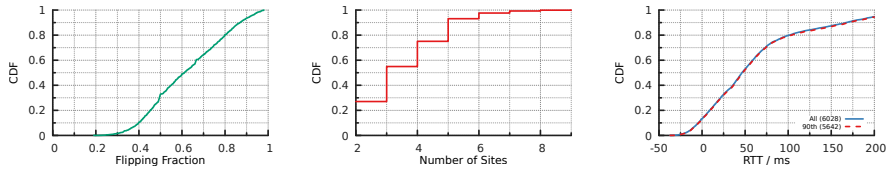
The anycast CDN's DNS returns 2 IP anycast addresses in resolutions of our test hostname. Therefore, we aim to measure anycast flipping for each of these IP addresses, separately. With the help of BrightData's technical support, we append 'dns-remote-dns-peer-info' in the proxy's user name string in the HTTP request, which instructs the proxy nodes to return the target IP address in an HTTP header of the response.

When a proxy node receives the response from a CDN server, it records a performance timing profile that records the timing information including connect time of each request and attaches the profile to the response. We use this information to measure the RTT from the proxy node to each anycast site it reaches. We describe more detail in § 4.4.

By default, BrightData may use a different proxy node per HTTP request. Since we wish to observe flipping per proxy node, this behavior is not desirable. Fortunately, we found a mechanism supported by BrightData to enforce a consistent session via the same proxy node for the current webpage we fetch [13]. To restrict all requests for fetching the mock webpage to use the same proxy node, we generate a random consistency token and concatenate it with our username registered with BrightData when we send the requests to the proxy service. With this mechanism, BrightData will use the same proxy node to send all the resource requests sharing the same consistency token. Further, BrightData allows a user to specify a geographic area to restrict the proxy selection. If an area is not provided, BrightData selects a proxy globally. Since we wish to study the impact of anycast flipping from as many vantage points as possible, we do not restrict proxy selection.

Although we use the consistency mechanism to ensure that BrightData does not switch proxy nodes in the middle of a mock page download session, we wish to validate that the mechanism is working as expected. We use the hashed IP field ('x-luminati-ip') in the responses to verify whether all 50 requests to the anycast CDN go through the same proxy node during the download session. We also made two additional requests to http://lumtest.com/myip.json with each download session to retrieve the proxy's public information (e.g., IP, AS number, geographical location). With this method, we found that among all 17,600 download sessions collected from May 31 to June 26 of 2024: 547 (3.1%) sessions did not succeed due to platform/proxy error; 713 (4.1%) sessions were issued from multiple proxy nodes (as many as 31 proxy nodes), even though we enforced the consistency token. For the remaining 16,340 download sessions, the same proxy node was used for all requests, but 186 (1.1%) sessions did not successfully complete all 50 requests. As we mentioned above, we also used requests to http://lumtest.com/myip.json to enrich the proxy's information, when the hashed IP matches the hashed IP in all 50 requests of the download session. For 72 more download sessions, the hashed IP in the requests to http://lumtest.com/myip.json did not match the hashed IP in the download session, so we excluded them. In the end, we continue with 16,082 measurements (91% of all measurements) that have a complete resource download session, and valid IP information retrieved from the the same proxy node.

To study anycast flipping, we also need to detect which site responds to each HTTP request. Fortunately, the anycast CDN we use adds a header field in its responses that contains an IATA code to indicate which site responded to an HTTP request. Therefore, if we observe responses containing different IATA codes are re-



(a) *Fraction of requests reaching the mode site* (b) *# sites reached per proxy node* (c) *ΔRTT of proxy nodes that flip between sites*

Fig. 7: *Anycast flipping results to an anycast CDN measured using BrightData. The* `All` *line in the rightmost figure includes all proxy nodes that experience flipping, and the* `90th` *line includes the proxy nodes that reach their mode sites less than 90% of the time.*

ceived by the same proxy node in response to our 50 requests, then we consider that the proxy node experiences anycast flipping.

In total, we discover 9,445 unique proxy nodes from 1,902 different ASes that can measure the RTTs from proxy node to the anyast servers. In the following section, we use the measurements through these proxy nodes to measure the prevalence of anycast flipping with the anycast CDN.

4.3 Prevalence

We now present the results on how frequently we observe anycast flipping to the anycast CDN. Out of 9,445 unique proxy nodes with timing profile, we observe 18,294 unique <proxy, anycast IP> pairs, since a proxy node can select different anycast IPs in their download sessions of retrieving 50 different resources. Among the unique <proxy, anycast IP> pairs, 6,028 (33.0%) pairs reach multiple sites for their requests, i.e., experiencing anycast flipping.

Figure 7a shows a CDF of the fraction of the multiple requests that are responded to by the site that receives the largest number of requests (We define such a site as the "mode site"). A value close to 1 means that reaching a different site is very rare, which is not indicative of anycast flipping. So, we set a threshold ≤ 0.9 to focus on the proxy nodes that experience frequent anycast flipping. This threshold leaves out 386 rarely flipping <proxy, anycast IP> pairs.

We show the CDF of the number of sites the remaining 5,642 <proxy, anycast IP> pairs reach in Figure 7b. As we can see, flipping between more than two sites is common, with 4,117 (68.3%) proxy nodes reaching three sites or more. In the extreme case, one proxy node flips between 9 different sites while downloading the 50 objects during our download session.

4.4 Impact on RTTs

Next, we investigate the impact of anycast flipping on the RTT from a client to an anycast CDN site. From the timing profile included in a BrightData HTTP response,

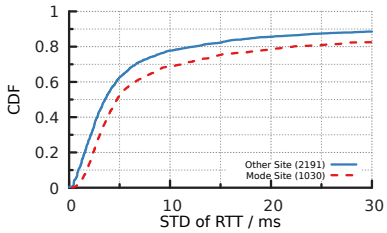


Fig. 8: *CDF of standard deviation of RTT to individual sites for proxies that experience frequent flipping (> 50%) and a large increase in latency on flipping (> 50ms). Minimum of 3 measurements per site.*

we extract the connect time – from when a proxy node sends a TCP SYN packet to the anycast CDN to when the proxy node receives the SYN+ACK reply. This metric estimates the RTT from the proxy node to its catchment site of the CDN, independent of where we run our curl script. A similar method is also used in [15].

We calculate the difference in RTT for each <proxy, anycast IP> as follows. First, we calculate the median RTT, mRTT, per site. Next, we determine if the site that receives the largest number of requests, site_x, has the largest mRTT. If it does, then we find the minimum mRTT to any other site and subtract site_x mRTT from it to determine the best case improvement in RTT that anycast flipping causes. If site_x does not have the largest mRTT, then we find the maximum mRTT to any other site and subtract site_x mRTT from it to determine the worst case additional RTT that anycast flipping causes. Figure 7c shows the CDF of these differences in RTT. Anycast flipping reduces RTT for less than 725 (13.0%) <proxy, anycast IP> pairs, typically by small values. Similarly, other proxies see modest increases in median RTT due to flipping, but 2682 (47.5%) <proxy, anycast IP> pairs suffer from RTT differences that are larger than 50 ms, and 747 (13.3%) <proxy, anycast IP> pair suffer from RTT differences larger than 150 ms. Among those 5,642 **90th** flipping pairs, 1,988 pairs flip **frequently**, mode site fraction is ≤ 0.5. And combined with our observations on flipping rates, 1,030 (5.6% of all) <proxy, anycast IP> pairs which mode site fraction is ≤ 0.5 and have an RTT difference $\geq$50 ms. In the next section, we investigate what implications the RTT increase has on web performance.

5 Impact on Web Performance

The previous section demonstrates the impact that anycast flipping has on the RTTs between proxies and the anycast CDN. We now turn to how these RTT increases may impact web performance by emulating a client downloading an existing webpage and flipping between two servers, focusing on clients that experience high flipping rates and large increases in latency. In particular, in emulating web performance we set the flipping rate(i.e., the client's chance of downloading webpage resource from a closer/faster server.) for TCP connections to 50% and the increased latency when flipping to either 10ms or 50ms. These rates and latencies are experienced by 8.9% and 5.6% of the BrightData proxies, respectively. While these percentages are low, CDNs serve

hundreds of millions of clients, so in absolute terms the number of affected clients may be quite large. Utilizing Mahimahi [33] measurements and emulations across 3 vantage points and 20 web sites, our experiments demonstrate that for a 50ms flipping latency penalty, the median increase in the First Contentful Paint metric ranges from 20.7% to 52.6% for HTTP/1.1 browsers and from 18.3% to 46.6% for HTTP/2 browsers. These results suggest for many clients the impact of flipping on web performance is significant.

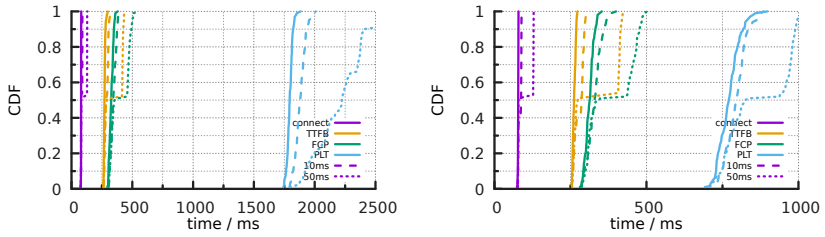
We recognize that simply adding a fixed latency penalty whenever flipping occurs assumes that the latency to the most common anycast site would have remained the same had flipping not occurred. We do not have direct experimental evidence to support this assumption, as we did not conduct experiments in which anycast packets were sent by a single proxy to multiple anycast sites simultaneously. Nevertheless, there is reason to believe that the differences in latencies to anycast sites are primarily due to unvarying causes such as the latencies of the (uncongested) network links on the paths to the sites. In particular, the standard deviations of the latencies are small compared to the large latency increases seen by some proxies on flipping, suggesting that variable causes of delay, such as network congestion and server load, play a more minor role. For example, as Figure 8 shows, among the <proxy, anycast IP> pairs that experience a mode site fraction $\leq$ than 0.5 and a median latency increase greater than 50 ms on flipping, 77.7% of the site landings' RTTs have a latency standard deviation under 10 ms when they flip to any particular site other than the most common. For the same group of <proxy, anycast IP> pairs, 68.7% of the site landings' RTTs have a standard deviation under 10 ms when they land on the mode site. In both cases, the statistics are quoted for sites with at least 3 measurements.

5.1 Infrastructure

To conduct our measurements we extend the web emulator Mahimahi. Mahimahi retrieves webpages, capturing all the resources – from different hosts – embedded in the webpage, and stores them locally. It enables replaying a webpage to a browser by serving the resources from Apache web servers, matching the resource and host associations collected during the capture. Optionally, Mahimahi can also add a synthetic delay during the replay. Thus, Mahimahi can emulate the web performance of a page load under varying network conditions. To study anycast flipping, we make several extensions to Mahimahi.

Per-host delay While Mahimahi allows a configurable delay to be applied to fetching resources during webpage replay, it applies the same delay to all fetches. This simplification is not faithful to the original webpage retrieval (webpages typically include resources from diverse hosts), nor is it convenient for measuring the impact of anycast flipping. So, we add the ability to set different synthetic delays per Apache web server to Mahimahi.

Support for anycast flipping The previous extension enables setting a delay per host, but with anycast flipping the delay to a specific host may vary depending upon the site reached. So we further extend Mahimahi to probabilistically apply additional delay per TCP flow to emulate flows reaching either a near site (i.e., no additional delay) or a far site (i.e., additional delay).



(a) *HTTP/1.1* (b) *HTTP/2*

Fig. 9: *Performance Impact of Anycast Flipping on the Mock Webpage*

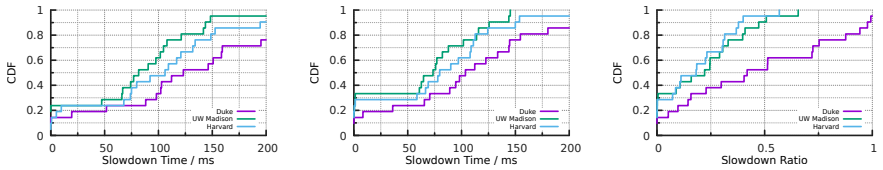
Support for HTTP/2 Mahimahi only supports HTTP/1.1 replay, yet HTTP/2 is frequently used today. Further, because anycast flipping operates on individual TCP flows and HTTP/2 handles multiplexing over the underlying transport differently than HTTP/1.1, we anticipate differences in the impact to each protocol. So, we update Mahimahi's Apache web server to support HTTP/2 by replacing the MPM prefork module ("mpm_prefork_module"), which does not support HTTP/2, with the MPM event module ("mpm_event_module"), and by adding HTTP/2 support with the "mod_http2" module.

Revised Retrieval Step Finally, we re-implement Mahimahi's retrieval mechanism This was necessary for two reasons. First, Mahimahi's retrieval does not support HTTP/2. Second, we need to collect the per-host delays used during emulation. To collect these delays we use 'tshark' to capture TCP packets on port 443. After retrieval, we extract the RTT to each host from the packet capture and apply the RTTs to Mahimahi's Apache web servers.

In emulation, our client is the headless Chrome browser in incognito mode, wrapped with Browsertime [4] to automate fetching webpages from Mahimahi and collect performance metrics.

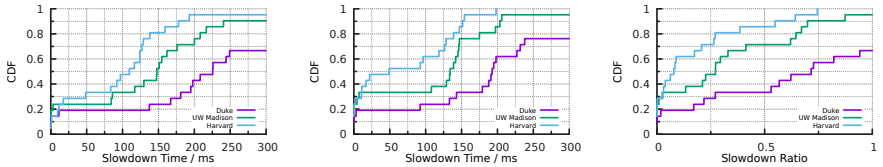
5.2 Mock Webpage Results

To show the impact of anycast flipping, we emulate web browsing of our mock webpage using Mahimahi. First, we retrieve the mock webpage from our vantage point at Duke University using our revised retrieval step. Because all of the one-pixel images in our mock webpage are served by the CDN, a single host serves the entire webpage. During retrieval, we calculate the RTT to the CDN from our vantage point and match the delay in emulation to the RTT. We then use our instrumented browser to fetch the mock webpage from Mahimahi 100 times. Figure 9 in the solid lines shows the CDF across the 100 measurements of four key performance metrics: connect - the TCP connection time for the webpage root object, TTFB - time to first byte of the root object, FCP [5] - first contentful paint is visible on the screen, and PLT - the start of the JavaScript load event. Smaller values in these metrics mean that



(a) *HTTP/1.1 Slowdown Time* (b) *HTTP/2 Slowdown Time* (c) *HTTP/2 Slowdown Ratio*

Fig. 10: *Slowdown in Average First Contentful Paint on Top-20 webpages*



(a) *HTTP/1.1 Slowdown Time* (b) *HTTP/2 Slowdown Time* (c) *HTTP/2 Slowdown Ratio*

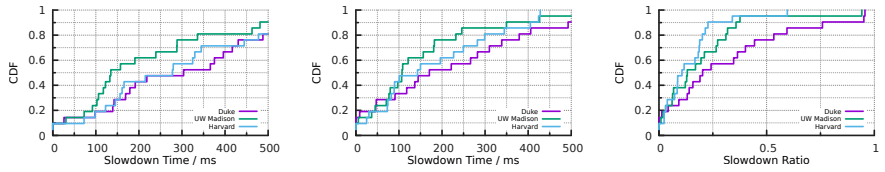
Fig. 11: *Slowdown in 90th Percentile First Contentful Paint on Top-20 webpages*

page loads more quickly and the web performance is better. We show the results for both HTTP/1.1 (Figure 9a) and HTTP/2 (Figure 9b).

Using the solid lines as our baseline, we next investigate the impact of two anycast flipping scenarios by selecting the single host serving our entire mock webpage to flip. To emulate anycast flipping observed in Section 4 that effects 6.3% of BrightData proxies, we set the probability of flipping to 50% and the additional RTT on a flip to 50ms — the dotted lines. We also run our emulations reducing the additional RTT to 10ms (which effects 9.7% of proxies) to see the impact of flipping when the RTT difference is less significant – the dashed lines. The metric values grow (become worse) as anycast flipping latency is added, regardless of protocol.

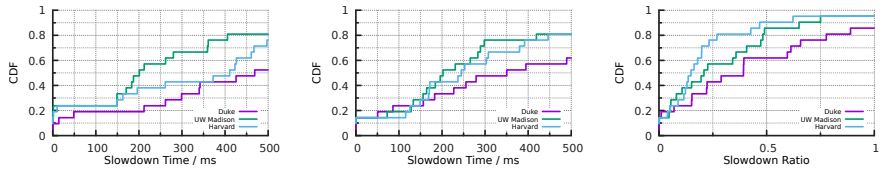
Consider the connected lines first. There is a noticeable step at roughly the middle of the distribution where flipping delay is added. Because flipping occurs per-flow and the flipping probability in our emulation is 50%, there is a 50% chance that the connection for the root object will reach the near site and the distributions in all three scenarios will be the same. However, if the connection for the root object is "unlucky", then the connect time is increased by 10/50ms. This applies to both protocols.

Unlike HTTP/1.1, however, Chrome's implementation of HTTP/2 uses a single TCP connection per host and multiplexes all requests over the same connection. If the first connection for the root object is unlucky, then Chrome remains unlucky for the rest of the webpage fetch. This is evident in the steps visible in all other metrics. By comparison, Chrome's HTTP/1.1 implementation uses 6 TCP connections in total per host, giving it six opportunities to reach the near site. Moreover, we observe that Chrome dispatches HTTP requests on the first available connection, so the connections to the near site receive a disproportionate number of the 81 requests because each request completes faster than other requests on connections to the far site. As a result, HTTP/1.1 performance under anycast flipping degrades slower then HTTP/2. This is



(a) *HTTP/1.1 Slowdown Time* (b) *HTTP/2 Slowdown Time* (c) *HTTP/2 Slowdown Ratio*

Fig. 12: *Slowdown in Average Page Load Time on Top-20 webpages*



(a) *HTTP/1.1 Slowdown Time* (b) *HTTP/2 Slowdown Time* (c) *HTTP/2 Slowdown Ratio*

Fig. 13: *Slowdown in 90th Percentile Page Load Time on Top-20 webpages*

most visible in the PLT dotted lines where HTTP/2 performance becomes significantly worse when the single TCP connect flips to the far site, while HTTP/1.1 performance more gradually degrades in steps as each of the 6 TCP connections flips to the far site.

We note that HTTP/2 performance overall remains better than HTTP/1.1. Even though HTTP/1.1 establishes 6 TCP connections, the 80 HTTP requests for one-pixel images are still transmitted serially across the connections. Meanwhile, because HTTP/2 removes head-of-line blocking, the 80 HTTP requests occur in parallel.

5.3 Popular Website Results

The mock webpage provides us with a synthetic view of the impact of anycast flipping, but real webpages typically aren't composed of 80 one-pixel images. So, next we explore the impact of anycast flipping on real webpages. In this analysis, we use the landing webpages of the top 20 most popular websites (Table A.2 in Appendix A) from the Tranco list [37]. Also, in addition to our vantage point at Duke University, we use two vantage points (Harvard University and the University of Wisconsin-Madison) provided by CloudLab [18] to retrieve the 20 webpages, giving us three distinct environments to measure the impact of anycast flipping.

As with the mock website, we apply a 50% flipping probability and add 50ms on flips to the far site. However, unlike the mock website, real webpages aren't typically served by a single host. Indeed, we observe that the 20 webpages we retrieve are served by a variety of hosts, including several CDNs. To study the impact of anycast flipping, we take the approach of assuming each of the CDNs uses anycast and emulate flipping to their hosts.

For each webpage, we run the Mahimahi emulation 100 times without flipping and then 100 times with flipping. We then compute the average and 90th percentile

for FCP and PLT (We do not present connect and TTFB here as the impact of these two metrics on web performance are also reflected on FCP and PLT). To ease visualization and comparison, we introduce two new metrics. Slowdown time is the difference between the same metric (e.g., average FCP) with anycast flipping and without anycast flipping. Slowdown ratio is slowdown time divided by the metric without anycast flipping. A slowdown ratio of 1.0 means that anycast flipping doubled the value of the underlying metric.

Figures 10a and 10b show the slowdown time for average FCP as a CDF across the 20 webpages for HTTP/1.1 and HTTP/2, respectively. For half of the webpages we study, the anycast flipping we emulate increases average FCP by 81-111ms for HTTP/1.1 and 77-108ms for HTTP/2, depending upon vantage point. Similarly, Figures 11a and 11b show the equivalent results for the 90^{th} percentile FCP and slowdown times over several hundred milliseconds are common. This logically follows from our results with the mock webpage, where we observe anycast flipping's impact is more significant in the tail of the distribution.

Interestingly, the results are similar regardless of protocol used, while on the mock webpage there was a notable difference. Not surprisingly, we find that hosting many objects on the same host is rare in real webpages. In fact, domain sharding – the process of splitting resources among many domain names to work around HTTP/1.1 head-of-line blocking by using additional parallel connections – is commonly used even with HTTP/2 [40]. Likely unintentionally, domain sharding currently mitigates that worst-case impact of anycast flipping on HTTP/2. Thus, website operators should consider the impact of anycast flipping on their websites before removing domain sharding.

Since there is no major difference in the results by protocol, we show the slowdown ratio of only HTTP/2 in Figures 10c and 11c for average and 90^{th} percentile FCP, respectively. The results vary by vantage point, which is to be expected as the RTTs to the hosts serving the webpages differ among the sites. Thus, the additional 50ms from flipping will have more/less impact. For Harvard and University of Wisconsin-Madison, most sites have small slowdown ratios, indicating that anycast flipping from those vantage points often does not significantly degrade perceived performance. Duke University, on the other hand, has significantly larger slowdown ratios, showing the time added by anycast flipping is a larger portion of the overall FCP. However, turning to the 90^{th} percentile for Harvard and University of Wisconsin-Madison, 8 webpages have slowdown ratios of about 0.5 or more, So, again, tail performance can be dramatically worse due to anycast flipping.

In Figures 12 and 13, the results for PLT time are shown, and are similar. In summary, for several of the webpages studied, anycast flipping – at levels observed to impact 1.3% of BrightData residential proxies – can significantly impact web performance.

6 Related Work

Measuring IP anycast performance. IP anycast has long been used by Internet services to provide automatic load balancing and latency reduction with multiple

anycast sites [14,11,34]. Many related works focus on measuring and analyzing the performance of existing IP anycast systems, including DNS root servers [27,30,32,29,21,25] and CDNs [14,48,25]. The main results from these studies are consistent: Global IP anycast does not always route clients to the sites that provide the lowest latency and does not always evenly distribute the workload among the sites [41].

Many previous works assume that each client reaches only one anycast site, which we observe is not true in practice. In our work, we show that, as many as 10% of the clients reach multiple different anycast sites due to anycast flipping. Schomp and Al-Dalky [42] also observe that 17% of clients flip between sites in proprietary DNS server logs, and speculate that one cause may be load balancers.

Load-balancer detection. From the examples in Section 2, load-balancing within the network is one cause of anycast flipping. Load-balancers are a widely deployed technique to utilize multiple links in order to support large traffic volumes. Paxson's measurement work on the diversity of routing behavior [36] is the first work to show the impact on the performance due to the load balancing, but only limited to unicast routing. Almeida et al. [8] proposed MCA (Multipath Classification Algorithm) to figure out the field in the packet used for load-balancing along a path, which in the presence of anycast can cause different flows – with different field values – to reach different sites. Paris traceroute [9] is a tool to perform path traces in the presence of load balancers. By keeping the fields load balancers might use constant, Paris traceroute tracks the complete path along one branch of the load balancer at a time. None of these works, however, consider the impact on anycast by load-balancers. Based on Paris traceroute, Augustin et al. proposed a multipath detection algorithm to probe the multiple paths of a load balancer [10]. Furthermore, Diamond-Miner [46] combines this multipath detection algorithm [10] with high-speed randomized probing techniques [12] to construct the Internet-scale topology with multiple paths.

Lan and Heidemann's work [49] is the first work that quantifies anycast flipping. In our work, we reproduced their results from 2016 with the same method, and update their results with a longitudinal study from 2016 to 2023. We find that anycast flipping is more common now, and extend our study to measure the performance impact of anycast flipping.

7 Future Work

This paper leaves open a number of interesting directions for future work.

Limitations. The web browsing performance study reported in this paper has several limitations. Most prominently, it's likely that important applications other than web browsing are also impacted by anycast flipping, and should be evaluated. Putting aside other applications, the browsing analysis measures only the First Contentful Paint and Page Load Time metrics. Metrics such as Connect Time, Time to First Byte, and even customer conversion rate could also be considered. Perhaps more importantly, the clients from which the popular web sites were downloaded were not very diverse. For example, they did not include hosts on home networks or mobile hosts using cellular data connections. Finally, we only emulated two fixed flipping latency penalties

(10 ms and 50 ms), both for the same flipping probability, 50%, which is a large flipping probability experienced by a relatively small fraction of clients. These values were selected based upon measurements of a single major anycast CDN. Yet, we know from our measurements of root DNS servers that anycast flipping impacts users of other anycast networks differently. Therefore, a more exhaustive study would evaluate a wider range of probabilities and penalties, selected from a larger set of anycast CDNs. Finally, our modeling of the flipping latency penalty assumes that an increase in the measured RTT is due to the change in routing to a different site, rather than the effect of a load balancing mechanism employed by the anycast CDN, which might apply to all sites. Whether such an assumption is warranted deserves more investigation.

Developing techniques to detect anycast flipping and determine the cause. In the cases of the root DNS servers and the anycast CDN studied in this paper the anycast site to which a client connects can be identified by techniques specific to these services, i.e., through the DNS CHAOS records provided by the root servers and the HTTP headers provided by the CDN. In general, however, it may be difficult for a client to determine even that the IP address that it connects to is an anycast address. Even more challenging is to determine why anycast flipping occurs when it does. There are a variety of possible reasons for flipping, including BGP and IGP route changes, IGP Equal-Cost Multipath (ECMP), and BGP Multipath. Anycast CDNs may even implement bespoke mechanisms to balance load among their anycast sites. While simple techniques like traceroute may identify route changes, the cause of the change is typically opaque to the client. Research on the prevalence and impact of anycast flipping would be enabled if there were more general techniques for detecting flipping and its causes.

Impact on anycast CDN cache performance. If a browser requests the objects embedded on a web page from multiple anycast CDN sites chosen essentially at random, then in order to guarantee cache hits the CDN must store all of the objects at all of the sites. Alternatively, anycast flipping may reduce the cache hit rate. Our experiments with the anycast CDN were oblivious to this effect, as we measured only TCP connect time, and not object download time. But it would be interesting to further explore the effect of anycast flipping on caching.

Preventing flipping. If it is deemed desirable to reduce the occurrence of anycast flipping, a number of potential approaches come to mind. First, at present Internet routers are generally oblivious to the notion of anycast addresses. If designated prefixes were reserved for anycast use, routers could disable load balancing for any datagrams or flows to those prefixes. Alternatively, perhaps datagrams could include "do-not-load-balance" tags. Second, anycast CDNs could employ a hybrid approach to delivering content, using an anycast address for the server delivering an HTML document, but then using a unicast address (for the same server site) for any embedded objects.

Mitigating flipping. We observed that in the default Chrome implementation, in HTTP/1.1 multiple TCP connections are established to download the necessary content, and the connections that perform best are used to download more objects. This adaptive adjustment to anycast flipping mitigates the impact of anycast sites with poor performance.

8 Conclusions

In this paper, we revisited the anycast flipping problem and provided evidence that it is increasing in prevalence, and that for some clients it can significantly impact web-browsing performance.

First, in recreating the measurements from [49], we showed that anycast flipping, first observed in 2016, is still present in 2024 and has grown more prevalent. Next, to study the impact of anycast flipping on web browsing, we conducted a measurement study of a major anycast CDN, finding again that anycast flipping impacts a small but significant number of vantage points. Further, a quarter of the vantage points that suffer anycast flipping also experience a large increase of at least 50 ms when directed away from their most common site. Finally, we measure the impact that frequent flipping with a large latency penalty has on web-browsing performance through emulations using Mahimahi. We find that, for this small but non-negligible portion of clients, the impact of anycast flipping on web performance can be significant.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and our shepherd Marcel Flores for their helpful comments, and Ramesh K. Sitaraman for helpful suggestions. We sincerely thank the Bright Initiative [7] for supporting us with proxy access. This work was supported in part by the National Science Foundation under Award 2225448.

References

1. BGP Bestpath AS-Path Multipath-Relax (2018), https://community.cisco.com/t5/routing/bgp-bestpath-as-path-multipath-relax/td-p/3709661
2. BGP User Guide: multipath (Protocols BGP) (2023), https://www.juniper.net/documentation/us/en/software/junos/bgp/topics/ref/statement/multipath-edit-protocols-bgp.html
3. Bright data, online proxy and web scraping platform (2023), https://brightdata.com/
4. Browsertime (2023), https://github.com/sitespeedio/browsertime
5. First contentful paint (fcp) (2023), https://web.dev/fcp
6. root-servers.org (2023), http://www.root-servers.org
7. Bright initiative, collaborative data solutions (2024), https://brightinitiative.com
8. Almeida, R., Cunha, I., Teixeira, R., Veitch, D., Diot, C.: Classification of Load Balancing in the Internet. In: Proceedings of INFOCOM. pp. 1987–1996 (2020)
9. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding Traceroute Anomalies with Paris Traceroute. In: Proceedings of IMC. pp. 153–158. ACM (2006)
10. Augustin, B., Friedman, T., Teixeira, R.: Multipath Tracing with Paris Traceroute. In: Workshop on End-to-End Monitoring Techniques and Services. pp. 1–8. IEEE (2007)
11. Ballani, H., Francis, P., Ratnasamy, S.: A Measurement-Based Deployment Proposal for IP Anycast. In: Proceedings of IMC. pp. 231–244. ACM (2006)
12. Beverly, R.: Yarrp'ing the Internet: Randomized High-speed Active Topology Discovery. In: Proceedings of the 2016 Internet Measurement Conference. pp. 413–420. ACM (2016)

13. BrightData: Session IP persistence. https://help.brightdata.com/hc/en-us/articles/4413171447953-Session-IP-persistence (2023)
14. Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., Padhye, J.: Analyzing the Performance of an Anycast CDN. In: Proceedings of IMC. pp. 531–537. ACM (2015)
15. Chhabra, R., Murley, P., Kumar, D., Bailey, M., Wang, G.: Measuring DNS-over-HTTPS Performance Around the World. In: Proceedings of IMC. pp. 351–365. ACM (2021)
16. Conrad, D.R., Woolf, S.: Requirements for a Mechanism Identifying a Name Server Instance. RFC 4892 (Jun 2007). https://doi.org/10.17487/RFC4892, https://www.rfc-editor.org/info/rfc4892
17. 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 IMC. pp. 477–488. ACM (2017)
18. Duplyakin, D., Ricci, R., Maricq, A., Wong, G., Duerig, J., Eide, E., Stoller, L., Hibler, M., Johnson, D., Webb, K., Akella, A., Wang, K., Ricart, G., Landweber, L., Elliott, C., Zink, M., Cecchet, E., Kar, S., Mishra, P.: The design and operation of CloudLab. In: Proceedings of the USENIX Annual Technical Conference (ATC). pp. 1–14 (2019)
19. Ed., R.Y.E.L.T., Ed., S.H.: A Border Gateway Protocol 4 (BGP-4). RFC 4271, RFC Editor (2006)
20. for Europe, U.N.E.C.: UN/LOCODE Code List by Country and Territory | UNECE (2022), https://unece.org/trade/cefact/unlocode-code-list-country-and-territory
21. 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: Traffic Monitoring and Analysis workshop (TMA) (2016)
22. Gray, C., Mosig, C., Bush, R., Pelsser, C., Roughan, M., Schmidt, T.C., Wahlisch, M.: BGP Beacons, Network Tomography, and Bayesian Computation to Locate Route Flap Damping. In: Proceedings of IMC. pp. 492–505. ACM (2020)
23. (IATA), T.I.A.T.A.: IATA Airport Code (2022), https://www.iata.org/en/publications/directories/code-search/
24. ipinfo.io: IP Info. https://ipinfo.io/ (2023)
25. Koch, T., Li, K., Ardi, C., Katz-Bassett, E., Calder, M., Heidemann, J.: Anycast in Context: A Tale of Two Systems. In: Proceedings of SIGCOMM. pp. 398–417. ACM (2021)
26. Lapukhov, P., Tantsura, J.: Equal-Cost Multipath Considerations for BGP. Internet Draft (2023)
27. Lentz, M., Levin, D., Castonguay, J., Spring, N., Bhattacharjee, B.: D-mystifying the d-root address change. In: Proceedings of IMC. pp. 57–62. ACM (2013)
28. Levine, M., Lyon, B., Underwood, T.: TCP Anycast: Don't Believe the FUD - Operational experience with TCP and Anycast. NANOG 37 (2006)
29. Liang, J., Jiang, J., Duan, H., Li, K., Wu, J.: Measuring Query Latency of Top Level DNS Servers. In: Proceedings of PAM. pp. 145–154. Springer (2013)
30. Liu, Z., Huffaker, B., Fomenkov, M., Brownlee, N., et al.: Two Days in the Life of the DNS Anycast Root Servers. In: Proceedings of PAM. pp. 125–134. Springer (2007)
31. Metz, C.: IP Anycast Point-to-(any) Point Communication. IEEE Internet Computing **6**(2), 94–98 (2002)
32. Moura, G.C., Schmidt, R.d.O., Heidemann, J., de Vries, W.B., Muller, M., Wei, L., Hesselman, C.: Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In: Proceedings of IMC. pp. 255–270. ACM (2016)
33. Netravali, R., Sivaraman, A., Das, S., Goyal, A., Winstein, K., Mickens, J., Balakrishnan, H.: Mahimahi: Accurate Record-and-Replay for HTTP. In: Proceedings of ATC. pp. 417–429. USENIX Association (2015)
34. de Oliveira Schmidt, R., Heidemann, J., Kuipers, J.H.: Anycast Latency: How Many Sites Are Enough? In: Proceedings of PAM. pp. 188–200. Springer (2017)

35. Partridge, C., Mendez, T., Milliken, W.: Host Anycasting Service. RFC 1546, RFC Editor (1993)
36. Paxson, V.: End-to-end Routing Behavior in the Internet. SIGCOMM Computer Communication Review **26**(4), 25–38 (1996)
37. Pochat, V.L., Van Goethem, T., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In: Proceedings of NDSS. ISOC (2019)
38. Randall, A., Liu, E., Padmanabhan, R., Akiwate, G., Voelker, G.M., Savage, S., Schulman, A.: Home is Where the Hijacking is: Understanding DNS Interception by Residential Routers. In: Proceedings of IMC. pp. 390–397. ACM (2021)
39. RIPE: RIPE Built-in Measurements. https://atlas.ripe.net/docs/built-in-measurements/ (2023)
40. Sander, C., Blöcher, L., Wehrle, K., Rüth, J.: Sharding and HTTP/2 Connection Reuse Revisited: Why Are There Still Redundant Connections? In: Proceedings of IMC. pp. 292–301 (2021)
41. Sarat, S., Pappas, V., Terzis, A.: On the Use of Anycast in DNS. In: Proceedings of 15th International Conference on Computer Communications and Networks. pp. 71–78. IEEE (2006)
42. Schomp, K., Al-Dalky, R.: Partitioning the internet using anycast catchments. SIGCOMM Comput. Commun. Rev. **50**(4), 3–9 (oct 2020). https://doi.org/10.1145/3431832.3431834, https://doi.org/10.1145/3431832.3431834
43. Schomp, K., Bhardwaj, O., Kurdoglu, E., Muhaimen, M., Sitaraman, R.K.: Akamai DNS: Providing Authoritative Answers to the World's Queries. In: Proceedings of SIGCOMM. pp. 465–478. ACM (2020)
44. Staff, R.N.: Ripe Atlas: A Global Internet Measurement Network. Internet Protocol Journal **18**(3) (2015)
45. Technologies, A.: Prolexic Routed (2020), https://www.akamai.com/us/en/multimedia/documents/product-brief/prolexic-routed-product-brief.pdf
46. Vermeulen, K., Rohrer, J.P., Beverly, R., Fourmaux, O., Friedman, T.: Diamond-Miner: Comprehensive Discovery of the Internet's Topology Diamonds. In: Proceedings of NSDI. pp. 479–493. USENIX Association (2020)
47. Villamizar, C., Chandra, R., Govindan, D.R.: BGP Route Flap Damping. RFC 2439, RFC Editor (1998)
48. de Vries, W.B., Aljammāz, S., van Rijswijk-Deij, R.: Global-Scale Anycast Network Management with Verfploeter. In: Proceedings of Network Operations and Management Symposium (NOMS). pp. 1–9. IEEE (2020)
49. Wei, L., Heidemann, J.: Does Anycast Hang up on You? In: Proceedings of Network Traffic Measurement and Analysis Conference (TMA). pp. 1–9. IEEE (2017)
50. Wei, L., Heidemann, J.: Does Anycast Hang up on You (UDP and TCP)? IEEE Transactions on Network and Service Management (TNSM) **15**(2), 707–717 (2018)
51. Zhang, X., Sen, T., Zhang, Z., April, T., Chandrasekaran, B., Choffnes, D., Maggs, B.M., Shen, H., Sitaraman, R.K., Yang, X.: AnyOpt: Predicting and Optimizing IP Anycast Performance. In: Proceedings of SIGCOMM. pp. 447–462. ACM (2021)

A Appendix

A.1 Geocodes Used by Different Roots

Table 2: Naming schemes used by different operators

Root	Operator	Naming Scheme	Regular Expression
A	Verisign	IATA, UN/LOCODE	`(rootns|nnn1)-((\w{3})\d|\w{2}(\w{3}).*)`
B	USC-ISI	IATA	`b\d-(\w{3})`
C	Cogent	IATA	`(\w{3})\d\w\.c\.root-servers\.org`
D	UMD	City/Country Code	`(\w{4})\d\.droot\.maxgigapop\.net`
E	NASA Ames	IATA	`\w\d{2}\.(\w*)\.eroot`
F	ISC	IATA	`\w{3}\.cf\.f\.root-servers\.org`
G	DoD NIC	Other	`groot-(\w{4})-\d`
H	ARL	IATA	`\d{3}\.(\w{3})\.h\.root-servers\.org`
I	Netnode	IATA, Other	`s\d\.(\w{3})`
J	Verisign	IATA, UN/LOCODE	`(rootns|nnn1)-((\w{3})\d|\w{2}(\w{3}).*)`
K	RIPE NCC	City/Country Code	`ns\d.(\w{2}-\w{3})\.k\.ripe\.net`
L	ICANN	City/Country Code	`(\w{2}-\w{3})-\w{2}`
M	WIDE	IATA	`M-(\w{3})-.*`

A.2 Top 20 web sites

www.amazon.com
www.apple.com
www.azure.com
www.baidu.com
www.bilibili.com
www.bing.com
www.facebook.com
www.google.com
www.instagram.com
www.linkedin.com
www.live.com
www.microsoft.com
www.netflix.com
www.pinterest.com
www.qq.com
www.twitter.com
www.wikipedia.org
www.wordpress.org
www.yahoo.com
www.youtube.com