Taming Anycast in the Wild Internet

Stephen McQuistin
University of Glasgow
Glasgow, UK
sm@smcquistin.uk

Sree Priyanka Uppu
Verizon Digital Media Services
Los Angeles, CA, USA
sree.uppu@verizondigitalmedia.com

Marcel Flores
Verizon Digital Media Services
Los Angeles, CA, USA
mflores@verizondigitalmedia.com

ABSTRACT

Anycast is a popular tool for deploying global, widely available systems, including DNS infrastructure and content delivery networks (CDNs). The optimization of these networks often focuses on the deployment and management of anycast *sites*. However, such approaches fail to consider one of the primary configurations of a large anycast network: the set of networks that receive anycast announcements at each site (*i.e.*, an announcement configuration). Altering these configurations, even without the deployment of additional sites, can have profound impacts on both anycast site selection and round-trip times.

In this study, we explore the operation and optimization of anycast networks through the lens of deployments that have a large number of upstream service providers. We demonstrate that these many-provider anycast networks exhibit fundamentally different properties when interacting with the Internet, having a greater number of single AS hop paths and reduced dependency on each provider, compared with few-provider networks. We further examine the impact of announcement configuration changes, demonstrating that in nearly 30% of vantage point groups, round-trip time performance can be improved by more than 25%, solely by manipulating which providers receive anycast announcements. Finally, we propose DailyCatch, an empirical measurement methodology for testing and validating announcement configuration changes, and demonstrate its ability to influence user-experienced performance on a global anycast CDN.

CCS CONCEPTS

• **Networks → Network architectures**; **Network performance evaluation**; *Network experimentation*; *Network measurement*;

ACM Reference Format:
Stephen McQuistin, Sree Priyanka Uppu, and Marcel Flores. 2019. Taming Anycast in the Wild Internet. In *Internet Measurement Conference (IMC '19), October 21–23, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 14 pages.

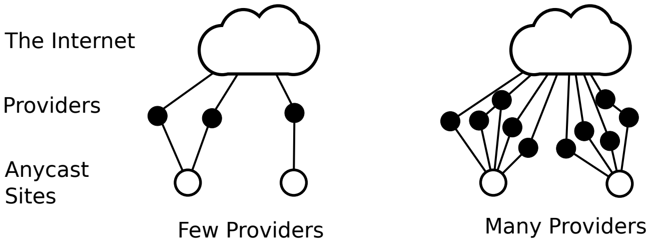


Figure 1: Anycast networks may feature either few (left) upstream network providers, or many (right).

1 INTRODUCTION

IP anycast is widely used for providing high-availability and low-latency network services, including DNS and CDNs. With IP anycast, network operators announce the same IP prefixes from multiple geographically-distributed sites. While each site provides the same service, the performance experienced by end-users can vary significantly based on the site selected and the path taken. However, BGP has no notion of latency or load, and may be heavily influenced by arbitrary network policy, further complicating the situation. Moreover, anycast networks do not have direct control over inbound routing: this is largely determined by the policies of upstream providers. Here, we specifically take *providers* to mean transit, paid, and exchange peers, as well as any network interconnection that provides client connectivity.

A significant portion of the prior work on anycast performance has focused on assessing the performance of anycast networks based on geographic distance, or the comparative performance of unicast addresses. However, such comparisons create unrealistic goals: in the absence of new meta-information or routing techniques (e.g., [27]), it is unlikely that such performance gains can be realized. Instead, we argue that the approaches with the greatest potential are those that can be deployed using the existing routing infrastructure.

We further argue that, in the case of many-provider networks with broad and diverse peering links, the correct mechanism for influencing routing decisions is through the announcements themselves, by altering the set of providers that receive them. However, doing so requires careful measurement to enable the attribution of observed changes.

In this paper, we demonstrate that anycast networks with more providers interact directly with a larger share of the Internet than networks with fewer providers. We show that having a large number of providers results in a greater proportion of short paths, with up to 86% of publicly visible paths consisting of only a single AS hop. We also demonstrate that such networks feature lower AS hegemony, a measure of the variety seen on inbound paths [19], showing the complexity and potential power of many-provider configurations.

Moreover, we examine how manipulating inbound routes alters the end-user performance of anycast. We demonstrate that these announcement changes may induce significant changes to both the site selected, as explored in previous work [15, 27], as well as changes in the route taken. 49.5% of vantage point groups shift site catchment (*i.e.* route to a different site), producing an over 25% reduction in round-trip time (RTT) for 30% of groups. However, adding or removing announcements to providers does not always result in *improved* RTT performance, and indeed indiscriminately announcing to new providers results in *reduced* performance in nearly 40% of networks.

To allow for the purposeful management of configurations, we present DailyCatch, a methodology for managing announcement configurations using routine empirical measurements. DailyCatch captures the performance changes induced by modifications to anycast announcement configurations, allowing operators to assess and weigh the impacts of any change. We further show, using measurements from a large, global, anycast CDN, that DailyCatch exposes a number of provider policies that confirm that managing provider configurations is highly non-trivial. While many large anycast networks have been studied previously, we believe we are the first study of the performance impacts of anycast configuration manipulation in a production, many-provider anycast network.

Previous studies have explored techniques for managing anycast networks, including proposals to use only a single provider [10], focusing on effective deployment at site level [15], working around poor anycast performance using DNS routing [12], and deploying new BGP communities [27]. While many of these approaches are effective in context, they may not be applicable in scenarios where many providers are necessary for scale and reliability, certain site choices are not available, significant rearchitecting is not possible, or upstream providers are unlikely to provide support for new communities. We therefore focus on the existing environment, in which having many providers is necessary and where many of those providers are non-cooperative. We further focus on solutions that can realistically be implemented in an existing, real-world network. To this end, we perform our measurements and analysis in the context of a global, commercial CDN that relies on anycast for its site selection. Despite our focus on anycast, many of our findings further apply to unicast deployments with similar many-provider arrangements.

We structure the remainder of this paper as follows. In Section 2, we provide an overview of anycast networks and structural insight into how such networks interact with the wider Internet at the BGP level. Next, in Section 3, we examine the specific impacts of announcement configuration changes on RTT performance. In Section 4, we present DailyCatch, and provide some examples of its measurements from the perspective of a global CDN provider. In Section 5, we examine some of the trade-offs of our design, including the challenges of performing active measurements. Finally, Section 6 describes related work and in Section 7, we conclude.

2 ANYCAST AND BGP

IP anycast is a technique for deploying distributed services. An operator announces the same IP prefixes from multiple physical locations, or *anycast sites.* Site selection is then performed by routers employing BGP in the process of routing traffic. The *catchment* of a given site is the set of clients that are routed to that site [28, 30]. Commonly deployed anycast networks consist of clusters of servers that service requests from end-users, *e.g.*, DNS root servers or HTTP servers in a CDN. In addition to the benefits of automatic site selection, anycast enables easy failover (*i.e.*, a site can go offline simply by withdrawing its routes, and BGP will determine new paths to alternative sites). In many cases, it also allows services to easily increase capacity by adding more sites, without having to scale up load balancing infrastructure. Finally, by deploying sites in diverse geographic locations with a diverse set of providers, operators can add significant robustness to their networks.

However, by pushing site selection into BGP, which has no notion of performance or load, anycast operators essentially cede control of their inbound traffic to upstream networks. Doing so exposes the anycast network to the impacts of upstream decisions: different transit providers may make different routing decisions and may apply different re-announcement policies. Other providers may re-announce only in specific regions or in other conditions. As described in previous work [15, 16, 27], these external decisions are difficult to predict, and they change not only *where* traffic arrives (*i.e.*, the site the traffic arrives at, and the catchment to which each user belongs), but also *how* traffic arrives at that site (*i.e.*, the AS path taken).

These challenges are further complicated by the need for *many-provider* anycast networks. While anycast networks often have complex networking configurations, consisting of both globally visible and local-network configurations [3], their specific use cases may determine their use of upstream providers. The frequent use of anycast for end-user facing services encourages operators to connect to a large number of networks [22]. This increased connectivity provides an opportunity for greater capacity (particularly in the case of CDNs) and greater reliability and fault-tolerance, balancing both provider and site failures. While previous work has demonstrated that using only a single provider may produce the most predictable results [10, 27], such a configuration is not feasible for large deployments, as noted in [27]. Indeed, directly connecting to networks offers significant potential to improve performance and reduce costs.

The complexity of configuring *who* receives anycast announcements, and *where* (*i.e.*, at which sites), however, creates opportunity for optimization. In having a large number of upstream providers, the anycast operator has a configuration space in which to manipulate their anycast configuration. By changing the set of providers that their anycast prefixes are announced to, operators may elicit a change in how a particular client reaches their network, either by influencing the site they are served by, or the path that their traffic takes to the same site. While we focus on anycast, our conclusions largely apply to unicast networks with many-providers.

In this paper, we investigate an approach for determining the impact that a particular set of announcements has on performance. First, however, we examine the observable structural differences between networks with relatively few providers, and those with more providers.

2.1 Choice of Anycast Networks

In order to develop an understanding of many-provider anycast, we examine multiple large anycast networks. As our first measurement target, we consider the DNS root servers. The DNS roots have been a common measurement target when studying anycast [15, 27]. They are appealing targets due to the relative availability of information on their deployments [3], as well as the fact that they are managed by multiple organizations with differing deployment strategies.

Next, we consider measurements from a large, globally deployed commercial content delivery network (CDN). This network features over one hundred sites spread across the world, with thousands of interconnects. Its large footprint and aggressive peering strategy provide an important viewpoint from an operational anycast network. From a BGP announcement perspective, the CDN operates multiple, independent anycast networks from the same AS. In particular, different announcements are restricted to particular physical locations in a region. In this study, we treat these as entirely separate entities, which we label CDN-1 through CDN-4. When serving end-users, DNS is used to map requests to the appropriate anycast network, providing a single global network. We note that ultimately this arrangement *dampens* some effects of many-provider networks, as each network has a limited scope.

Finally, we consider Google DNS (8.8.8.8, which we label GDNS) as it also represents a large, global, high-traffic anycast network.

Each of these networks is designed to service different traffic patterns, with the root servers providing service to DNS resolvers, the CDN servicing HTTP requests, and GDNS acting as a local resolver. We aim not to assess the performance of any particular approach, but to instead demonstrate that the differences between them create the opportunity for optimization based on announcement configurations.

2.2 Many-Provider Networks

In this section, we demonstrate that, fundamentally, large, many-provider networks interact more directly with a larger portion of the Internet than those that operate with only a few providers. These structural differences change the way in which anycast must be managed and increase the need for direct, evidence-based anycast management. This further emphasizes the need to study both the resulting catchments, as in previous work [15], and the need for determining *how* (*i.e.*, which path) clients take to the network.

We examine public BGP data from a set of RouteViews [4] collectors[1]. While BGP reveals many of the structural relationships that exist, it is important to note that the data used here may underestimate the effects that some of these relationships have. This is due to the nature of public BGP collectors and the presence of private peering links that may not be visible in public BGP data. However, such limitations are well known [32], and do not affect the generality of our conclusions.

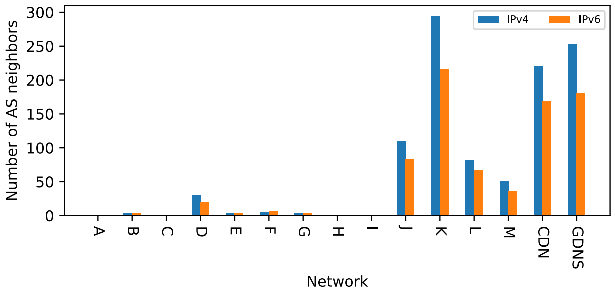


Figure 2: Counts of unique AS neighbors for DNS root servers (excl. G and H), the CDN, and GDNS.

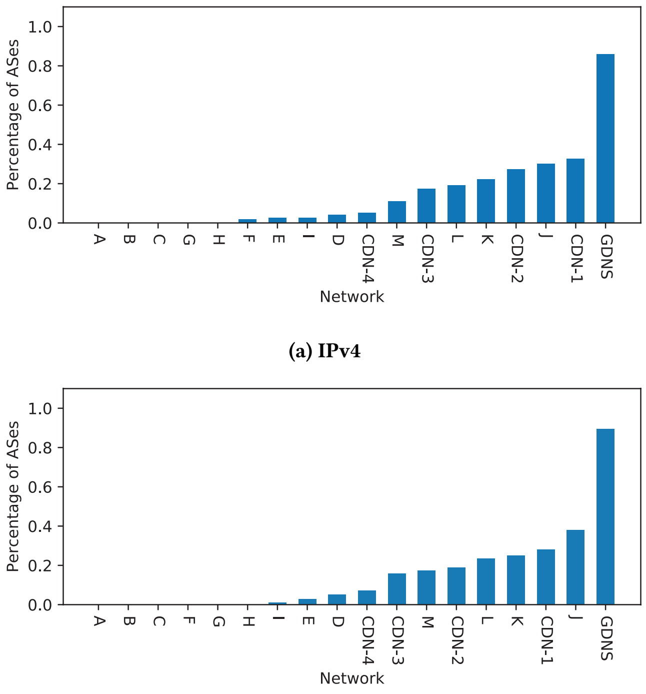
Neighbors Figure 2 shows counts of the unique AS neighbors[2] for each of the DNS roots, the CDN, and Google's DNS service. As shown, the aggressive peering policies of both the CDN and GDNS are apparent, as they have significantly more AS neighbors than all but the K root. This view presents us with two broad groups: those with *few* AS neighbors (fewer than 10 upstream providers), and those with *many* neighbors (10 or more providers). While neighbor counts alone do not determine routing behavior, they provide insight into how many possible routes a client may take to the anycast network. We acknowledge that the boundary between the few and many provider classes (of 10 providers) is somewhat arbitrary. However, the goal of this classification is not to determine a precise boundary between each class, but to use these classes to illustrate the differences in how networks with varying numbers of providers interact with the Internet.

There is an important difference between having more neighbors and more sites. Indeed, an anycast network may have many sites that all exist within a single, or handful of providers: for example, the E root has 234 sites, but announces to only 3 publicly visible providers. On the other hand, they may have relatively few sites with a large number of providers at each: the M root has 9 sites, but announces to 43 providers. We argue that these provider relationships, and which providers receive anycast BGP announcements, are another dimension in the anycast configuration space, comparable to the deployment of sites.

Single-Hop Paths Next, we consider the AS path lengths seen in the RouteViews BGP paths. Figure 3(a) considers the percentage of one-hop AS paths (when grouped by source AS number) that are seen from RouteViews, for the same set of targets as described above. As shown, the grouping (many vs. few providers) is evident in this dataset. We can see a loose correlation between neighbor count and the proportion of short paths: A, C, H, and I, which have only 1 provider, have no or very few one-hop paths, D, L, and M, which have tens of providers, have relatively more, and J, K, the CDN prefixes, and GDNS, which have hundreds of providers, have the greatest proportion of short paths in the dataset. This correlation is not absolute, being heavily dependent on the nature of the upstream providers, but it holds that, broadly, having more providers increases the proportion of one-hop AS paths observed.

[1] `route-views.{2, 3, 4, 6, eqix, isc, telxatl, nwax, sfmix, chicago, flix, kixp, jinx, linx, soxrs, napafrica, wide, sydney, sg, saopaulo, chile}`

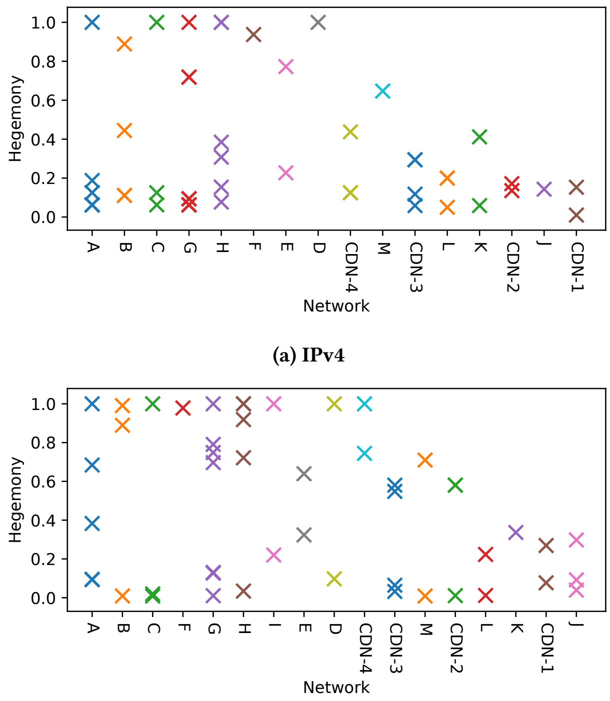
[2] Where announcements are multi-homed (*i.e.*, in roots A and J), neighbors are aggregated across each announcing AS.



(a) IPv4

(b) IPv6

Figure 3: Percentage of single-hop AS paths (grouped by source AS number) as seen from RouteViews for DNS root servers, the CDN, and GDNS.



(a) IPv4

(b) IPv6

Figure 4: Observed hegemony values for DNS root servers and the CDN. GDNS has no detectable provider hegemonies.

Figure 3(b) plots the same experiment, but with IPv6, where we see the conclusions hold.

We do not aim to make any claims related to the *performance* of such AS paths as a result of their length, but instead to show that the large number of providers results in differences in the nature of the paths used by clients. In particular, the shorter paths suggest many of these providers are topologically valuable, highlighting the potential impact of announcing (or withholding) an anycast address to each of the providers. An anycast configuration must take into consideration that announcing to providers may alter the catchments *or* the path taken to the same site.

Hegemony Next, we examine the interaction of these networks with the larger AS graph. To this end, we consider differences seen in the AS hegemony between each of the anycast networks. AS hegemony is a metric that measures the importance of upstream networks based on an improved version of betweenness centrality [20]. Hegemony provides a value from 0 to 1 which can be interpreted as the weighted fraction of paths that cross a given AS. Hegemony provides such an assessment regardless of whether or not a given AS is a direct neighbor. It is further possible for a network to have many neighbors which are often unused, resulting in high hegemony for few, or even a single, provider. Here, we compute the hegemony for the specific anycast *prefixes*, rather than the entire AS, to develop a better understanding of the specific reachability from the Internet. We make this adjustment as anycast operators may implement different announcement policies on a per-prefix basis, complicating AS-level results. We use the hegemony values to determine how dependent each network is on a single, or small number of, providers.

Figure 4 presents the observed hegemony values for each anycast network sorted by the number of one-hop AS paths. First, we note that GDNS does not appear, as it has no upstream networks with significant hegemony at all, which suggests that their paths are diverse, and they have no centralized dependencies. We see that, as with our AS neighbor analysis, there are two broad groups: the first features at least one upstream network with extremely high hegemony, indicating that a single provider lies on almost all observed paths. The many-provider prefixes, however, exhibit low hegemony values almost across the board, with only 1 many-provider network showing an AS with hegemony over .5 in the IPv4 case. The prefix for CDN-4 is available to fewer publicly visible providers, resulting in increased single-provider hegemony. We also see that the hegemony values are generally higher in the IPv6 case, which may be the result of greater dependency on a few major IPv6 providers, as well as a decrease in publicly visible links.

Given knowledge of the operational configuration of each of these networks, these results are not surprising. The C root, for example, has a single provider. Therefore, we see a hegemony of 1 for AS174. For the CDN networks, the largest hegemony values come from transit providers, which, even with significant peering, account for a large number of inbound paths.

AS	# Probes	# Catchments
A	307	11
B	45	7
C	30	5
D	24	6
E	31	6

Table 1: Top North American ASes with 2 or more vantage points, ranked by traffic volume.

Based on this data, we conclude that for most networks there exists a correlation between the number of AS neighbors (Figure 2), the proportion of short paths (Figure 3), and hegemony (Figure 4). Many-provider anycast networks are likely to have a significant portion of very short paths, and exhibit little to no AS hegemony. This correlation shows that many-provider networks must be analyzed and managed differently than their few-provider counterparts: their BGP-level interaction with the rest of the Internet is observably different, and their announcement configurations feature both a richer configuration space, by virtue of the increase in providers and a wider set of potential impacts. Furthermore, we see the root servers are split across the many-/few-provider divide, an important consideration in studies that rely on them.

3 NETWORK IMPACTS

Section 2 showed that many-provider networks interact with the Internet via a wider variety of links, creating significant potential for optimization. We emphasize that the presence of these links is a critical component of the operating model for many of these networks, offering greater capacity and resilience. These features are a requirement for many large anycast networks: the benefits are necessary for day-to-day operations.

In this section we examine the impacts of manipulating anycast announcement configurations by altering which providers receive announcements. We aim to understand: (1) *how* changes in configuration affect RTT performance; (2) how RTT changes when the configuration alters *where* and *how* clients connect; and (3) which *types* of networks are impacted.

3.1 Experimental Setup

We conduct a large-scale experiment on the anycast CDN network measured in the previous section. The experimental design is comprised of two large-scale conditions (*i.e.* anycast configurations). While many upstream networks on the Internet are not independent, this approach provides us with a series of examples that demonstrate the profound impact that announcement configurations can have.

As our first announcement configuration, we provide announcements to a restricted set of *transit providers*, on the order of 2 per site. These are service providers that re-announce to other networks, extending connectivity. This configuration represents an approximately minimal set of providers for full connectivity. Under the second configuration we provide announcements to the same transit providers as in the previous set and to nearly all available providers, including peers at Internet Exchange Points (IXPs) and private interconnects, on the order of hundreds per site. This configuration offers an implementation of a many-provider network that takes advantage of local connections where possible. Each of these configurations is simultaneously implemented using experimental announcements at the CDN, making use of all 4 CDN networks described in the previous section. The announcements are made using separate anycast blocks specifically configured by the CDN for the purposes of regular testing and configuration management, separate from customer operations.

We perform a series of active measurements, taking traceroutes from approximately 10,000 RIPE Atlas [2] probes around the world. All measurements were taken during a single day in April 2019 and were directed at CDN DNS names pointing at each of the above configurations[3]. We further collect data from a set of passive measurements taken from beacon traffic collected at the CDN as a matter of course in CDN operations.

3.2 Grouping Vantage Points

To discuss meaningful changes in RTT performance, we group vantage points together. This grouping allows us to identify the broad path-level changes behind any performance differences. Ideally, vantage points (VPs) that *share fate* should be grouped together: group members should fall within the catchment of a single (but perhaps different across configurations) anycast site. As shown in Table 1, using RIPE Atlas probes as our vantage points and grouping by AS alone is insufficient: it groups together vantage points that map to different catchments. To improve the groupings, we investigate dividing vantage points into further sub-groups.

We consider four sub-grouping functions, each sub-dividing ASes by: (1) geolocation (country or US state); (2) prefix of the probe's public address, as reported by the RIPE Atlas platform; (3) prefix of the first public hop on a traceroute to a common unicast target; and (4) prefix of the last hop on the same unicast path. To measure the impact of a sub-grouping function, we consider two metrics, *similarity* and *coverage*, across groups with more than one vantage point. To measure overall similarity of a group, we use a generic vantage point similarity metric, which captures group similarity independently of the ultimate measurement target. To begin, we define the Jaccard similarity [24] of probe x with respect to another probe y as:

$$d_{(x,y)\to m} = \frac{|P_{x\to m} \cap P_{y\to m}|}{|P_{x\to m} \cup P_{y\to m}|}, \tag{1}$$

where m is a destination IP, and $P_{x\to m}$ is the path from x to m. As noted in [24], basing similarity upon paths to a single destination lacks robustness. To address this, we compare the similarity of paths to a set of destinations, M (where $m \in M$). Here, M is the set of all destinations within the RIPE Atlas probe's built-in measurement set (*i.e.*, those measurements that all probes routinely carry out). This provides a diverse set of destinations, reducing dependence on any one destination or routing configuration, without increasing probe measurement burden. We calculate the similarity for each probe, sim(x), within group g, as:

$$\text{sim}(x) = \underset{y \in g}{\text{median}}\{d_{(x,y)\to M}\} \tag{2}$$

[3] The endpoints used in these measurements, and their measurement IDs, are available upon request to research@verizondigitalmedia.com.

Next, we calculate a similarity for each group, g, as:

$$\mathrm{sim}(g) = \operatorname*{median}_{x \in g}\{\mathrm{sim}(x)\} \tag{3}$$

Finally, we define the similarity of each sub-grouping *function*, F, as:

$$\mathrm{sim}(F) = \sum_{g \in G} \mathrm{sim}(g) \tag{4}$$

where G contains groups produced by F, and singleton groups have been removed to limit the impact of anomalous probes. Medians are used throughout the similarity metric calculation to dampen the impact of outliers. Beyond similarity, we further compute the *coverage*. We define coverage as the number of ASes that have at least one group after a sub-grouping function has been applied and singleton groups removed. If a given group function creates too many singleton groups, we may lose coverage of ASes.

Our results show that sub-grouping by geolocation significantly increases similarity (+10.0%), with only a slight reduction in coverage (−2.2%), compared to grouping by AS alone. All of the other sub-grouping functions either significantly reduce coverage (probe prefix and first hop prefix), *or* do not meaningfully improve similarity (last hop prefix). To validate our choice of sub-grouping by geolocation, we again consider the ASes listed in Table 1. When sub-grouped by geolocation, each of the sub-groups listed fall into the catchment of a single anycast site. Looking beyond these, at *all* ASes, sub-grouping by geolocation results in 60% of groups falling within the catchment of a single site, and 90% of groups within the catchments of 2 or fewer sites, approximately achieving our goal of each group having shared fate.

3.3 Vantage Point Selection

In the course of our analysis we consider two primary sources of information. First, we examine RTTs measured on the server side via CDN logs. These measurements rely largely on connections from non-production real user measurement (RUM) beacons from a broad set of CDN clients. As a result, they can be manipulated for the purposes of these experiments and pointed at addresses implementing our configurations. In order to *reproducibly* geolocate these clients, we use the MaxMind geolocation database.

While they offer a broad set of real user measurements, these beacons are only able to provide us with passive end-to-end data and contain no information about paths. Path information is critical for providing a robust analysis of both catchments and inbound routing behaviors. Therefore, we need a further data source that is able to provide to-the-CDN traceroute information: in this case, we use RIPE Atlas.

In order to ensure that measurements from RIPE Atlas accurately portray the behavior of clients, we compare these two sets of vantage points. Here, we consider the *absolute* difference between the median beacon and median probe values for each group, as we are attempting to assess the accuracy of the probe-based measurements. Figure 5 presents a CDF of the RTT difference between the beacons and RIPE Atlas measurements, for all groups visible in both datasets that had at least 2 RIPE Atlas probes. We further filter out outlier beacon measurements, removing all results over 200ms. Here, we see that 82% of groups have less than 15ms of error, suggesting that the RIPE Atlas data provides a relatively accurate sense of client RTT performance. We present a further analysis in the coverage provided by RIPE Atlas in Section 5.1.

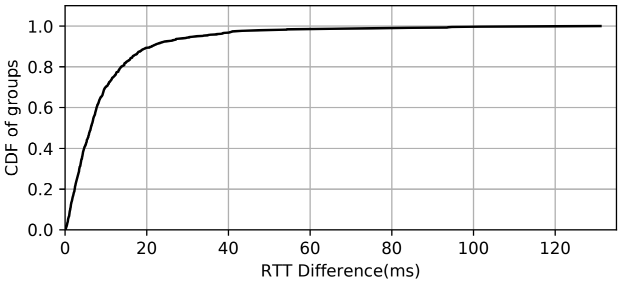


Figure 5: Changes in median RTT between the CDN beacons and RIPE Atlas probes.

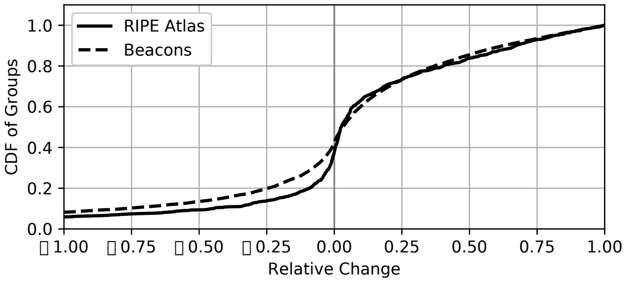


Figure 6: Changes in median RTT (between control and experiment configurations) for all groups.

3.4 Performance Impacts

Figure 6 presents a CDF of the median change in RTT between the transit only and all provider configurations, for both RIPE Atlas and the CDN beacons over all groups. In this figure, and the remainder of the section, we focus on the change in behavior of the median of each group. The key notion is to observe the overall behavior of the group, and how it changes as the result of different announcement configurations, rather than the RTT distribution within each group. This allows us to assess the impacts of changes at the group level, across all groups, regardless of size of their internal characteristics. We further consider the relative change observed in order to understand the impacts to each group individually. Doing so does not show the absolute changes. Instead, it provides a concise description of how the RTT changed from the perspective of each group. A positive value indicates that the RTT decreased in the experimental configuration, *i.e.* the addition of announcements to more networks presented alternative paths and performance improved. A negative value indicates a reduction in performance. We see that 60% of groups saw an improvement in performance with the expanded announcement configuration, while nearly 40% of groups saw a decrease, for both probes and beacons. This wide spread suggests that care must be taken: indiscriminately adding announcements can reduce performance.

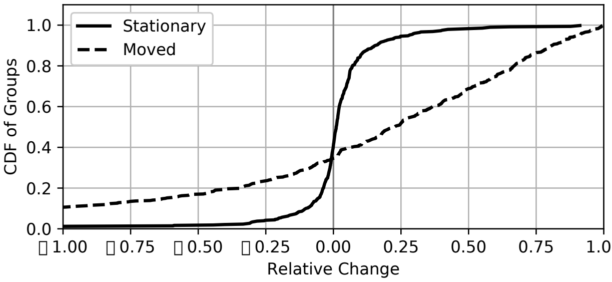


Figure 7: Groups which shifted catchment, and were therefore served by a geographically different location saw the biggest change.

Catchments However, examining all of the groups at once does not paint a complete picture. The interactions between upstream providers and the anycast network may result in changes beyond performance alone. To provide further insight into these interactions, we refine this view. First, we examine the difference between groups that changed catchment (*i.e.* the probes went to a different site), and those that remained stationary (*i.e.* the probes connect to the same site in configurations).

Figure 7 shows a CDF of the relative change in RTT for both the moved (468) and stationary (477) groups. As expected, the majority of differences, as well as the most extreme changes, come from groups that change catchments. This follows the intuition: end-to-end delay is often dominated by propagation delays. However, this was not uniform, as the stationary groups saw *some* (generally less than 25%) change in RTT. This shows us that while catchment is an important consideration, as seen in [15], the path itself also impacts RTT performance. These results further demonstrate that shorter paths need not result in improved performance. While the lack of such a relationship is well known, our findings here demonstrate this empirically.

Inbound Paths Next, we consider an analysis of the inbound paths taken by the probes in these measurements. Here, we map our traceroutes into AS paths using a RIB from RouteViews taken from the same day, revealing how each probe reached the CDN. For simplicity, we ignore hops that were not responsive (potentially underestimating AS path lengths). When comparing paths, we consider a group to have taken a *different* path if *any* probes traversed a different set of ASes. Here, we seek only a coarse grained sense of what a *different path* means. In particular, we are largely concerned only with the appearance or disappearance of providers, ignoring many of the more subtle components, including AS boundaries.

When we examine the catchments that remained stationary, we see that about 75% of these groups took a new AS path to the anycast destinations. Further subdividing, we found that of the groups that performed worse, only about 72% took a different path, but for groups that improved, 77% of the groups took a new path. These results suggest that taking a new path to the same destination does not necessarily mean that performance will improve. These findings reinforce the notion that there is no simple answer: new providers may create new BGP paths, but they need not result in better performance.

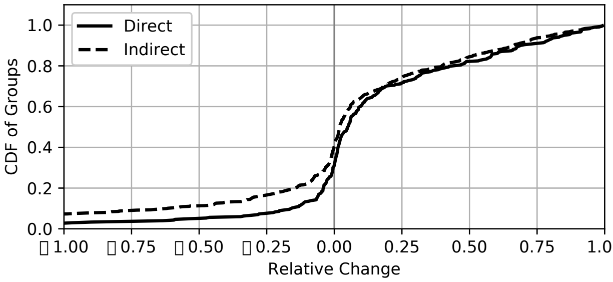


Figure 8: Groups which receive direct announcements, versus those with only indirect connections.

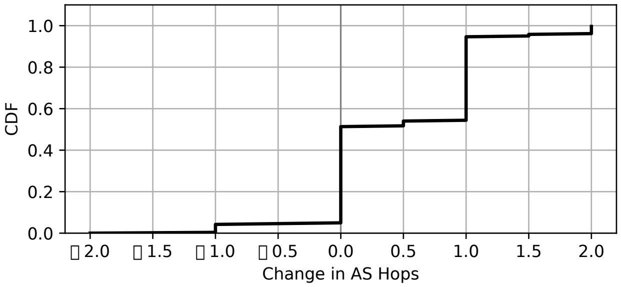


Figure 9: Of indirect-groups that got worse, 50% of the paths became shorter.

Providers Figure 8 presents the groups separated by the nature of the relationship with the probe's network: the solid line indicates direct peering with that network (304 groups) and the dashed line indicates networks with no direct announcement (640 groups). Surprisingly, for groups that performed better, the improvements are approximately the same: in many cases, downstream networks took advantage of the newly offered paths.

On the lower end of the distribution, nearly twice as many indirectly-connected networks saw a decrease in performance. Figure 9 shows the change in AS hop count by the indirect-groups that saw a performance decrease. Here, we take the median decrease for each group. Fractional changes indicate a mixed-behavior group. We see that 50% of these groups saw a *decrease* in path length. This suggests that while their median RTT increased, the new announcements provided a shorter AS path via a different provider. Previous work has found that peering links outperform their transit counterparts in many cases [5]. Our findings here do not necessarily contradict this, but serve as an indicator that managing the inbound paths to a network is a different issue than overall or egress performance.

Finally, to develop an understanding of the potential impact each individual provider has, we consider an analysis of the number of origin networks that we see using each inbound provider. Here, we examine the traceroutes and the penultimate hop seen before they arrive at the CDN network. We count the number of origin ASes that traverse each unique adjacent AS. We note that this includes regional providers, who provide transit to other networks, eyeball

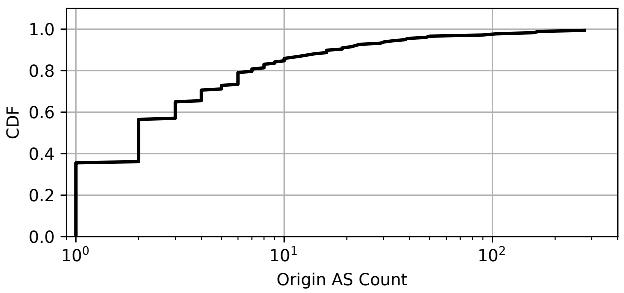


Figure 10: The number of ASes that arrived via each inbound network.

networks, IXPs, and other networks. For simplicity, we discard all traceroutes in which the penultimate hop was non-responsive, making these numbers an underestimate.

Figure 10 presents the results. We consider a set of 172 observed inbound providers, and exclude the transit-only providers. We see that about 40% of networks provided connectivity to only themselves. This is expected given the nature of the CDN network: it has a strong incentive to connect directly with eyeball providers, which are often leaves in the AS graph. The remaining 60% of networks, however, show the complexity of the situation: with many networks offering connectivity for a large number of networks, there are potentially significant and cascading impacts to their anycast behavior. Any system for anycast management built around many-provider relationships will have to account for this diversity, properly handling the potential for announcing to both leaf-networks with no dependencies, and transit providers of varying sizes.

The variety of paths taken, as well as the noted differences in performance, indicate the complexity of the configuration space: announcing to additional providers may increase route diversity, add capacity, improve availability, and alter site selection. It may, however, highly impact performance, both for direct neighbors, and for more distant upstream networks. Indeed, our analysis has demonstrated that there are significant gains to be had, with some networks seeing a nearly 99% improvement on RTT. However, these improvements are not uniformly realized: many networks saw a significant *decrease* in performance. Therefore, while managing anycast at the provider level can be extremely powerful, any proposed anycast configuration must be thoroughly tested for such far reaching impacts.

4 EVALUATING MANY-PROVIDER ANYCAST ANNOUNCEMENTS

To properly manage many-provider anycast configurations, it is necessary to empirically *compare* two configurations. We present DailyCatch, a methodology that uses active measures to provide such a primitive. DailyCatch conducts experiments by measuring a *control* anycast configuration and an *experiment* configuration. DailyCatch assesses the difference between these configurations, generating a *score* (*i.e.*, a numerical value distilling the latency difference) for each sub-AS group. These groups provide *context*, allowing for actionable information about *how* and *where* catchments and performance have changed between configurations. It is typical that some groups of vantage points (here, RIPE Atlas probes using the groupings described in Section 3.2) will see their performance improve, while others will be degraded: DailyCatch groups VPs together such that these results can be meaningful to network operators.

The basic measurement component of DailyCatch is a *snapshot.* A snapshot consists of a set of traceroutes taken from a large number of vantage points towards the same anycast target at the same time, along with the anycast catchments for each vantage point. Snapshots contain information about groups of vantage points, where groups contain vantage points that belong to the same AS and geolocation. In Section 3.2, we demonstrated this grouping function provided a good trade-off between coverage and similarity.

Section 4.1 describes the core contribution of DailyCatch: the mechanism by which two snapshots are compared. Our scoring methodology must be capable of surfacing changes in performance that are significant and relevant to the network being measured. To this end, we use information on traffic volumes and catchments observed by the global, commercial CDN used in the previous section. To do this, DailyCatch must scale changes in latency to ensure that it is capturing the most meaningful differences, while allowing scores to be compared. After describing the driving principles in the development of DailyCatch, we demonstrate its utility through a series of case studies in Section 4.2. While the inputs and configurations come from a CDN environment, many of these features, and the ultimate observed behaviors, are generic and may apply to many large, many-provider networks.

DailyCatch's scoring mechanism enables operators to evaluate the difference between two announcement configurations, in terms of client performance. There are two main ways in which operators can make use of DailyCatch's output to improve and manage their anycast announcement policies. First, DailyCatch produces a net score (described in Section 4.1) that captures the broad performance impact of a policy versus the control configuration. This net score indicates whether or not a given configuration should be adopted in its entirety. However, this is a fairly coarse measure: there may be clients that are significantly impacted by the configuration, even if those impacts are outweighed by improvements in performance for more important networks (*i.e.*, those that have larger footprint). The second use of DailyCatch is more nuanced: operators can inspect those vantage point groups whose scores shift most significantly, and make targeted adjustments to their announcement configurations. Section 4.2 describes how this approach works in practice, giving an example configuration experiment, and guidance on how operators can interpret the results. The path information that DailyCatch captures is crucial in allowing operators to identify the necessary configuration changes.

4.1 Scoring and Comparison

Snapshots capture the latency and catchment membership observed by probe groups at a given point in time, towards a particular target. However, the motivation for DailyCatch is to grant the visibility required to make evidence-based configuration changes. This means that we need to be able to *compare* two snapshots, each describing

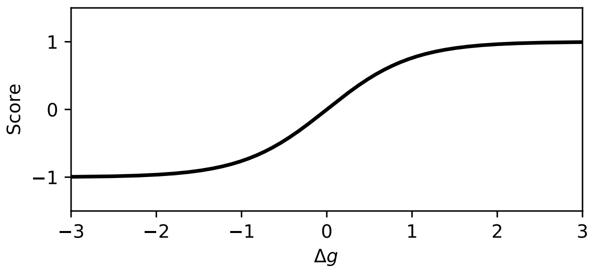


Figure 11: Rationale for logistic function: variation from a 100% **decrease to a** 100% **increase.**

a particular configuration, and highlight the differences between them. Central to this is a scoring function, allowing DailyCatch to quantify the difference between two snapshots at a per-group level. Our scoring function must surface *meaningful* and *relevant* performance changes. We use a logistic function to ensure that the score is meaningful, by dampening the impact of anomalous results. Finally, we weight each group's score by the relative importance of that group to the CDN, in terms of how much client traffic it represents.

To begin, we define the change in RTT *within* a group, g, between two snapshots, a and b, as:

$$\Delta g = \frac{rtt_a - rtt_b}{rtt_a} \quad (5)$$

where rtt_x is the median RTT of vantage points in snapshot x. Medians are used to remove the effects of anomalous measurements[4].

We normalise the RTT change with a logistic function:

$$s_{\text{raw}}(\Delta g) = \frac{2}{1 + e^{-k(\Delta g)}}. \quad (6)$$

The aim of the logistic function (Figure 11) is to ensure that we capture meaningful changes in RTT. Selecting an appropriate k value (*i.e.*, the steepness of the logistic curve) limits the impact of anomalous latency measurements. While we select $k = 2$, based on the range of differences seen in Section 3, which devotes the majority of the score to the space between a 100% decrease in performance and a 100% increase. Other k values can be used for more dramatic changes, in which more than 100% is needed. Applying the logistic function gives a value that varies from −1 (RTT degraded significantly) to 1 (RTT improved significantly).

Next, we weight the score for each group by its *relative volume*, v. The relative volume of an AS is the fraction of inbound traffic that originates from that AS. In the context of the CDN, we calculate relative volume by measuring the TCP SYN packets that are seen in a 24 hour period, sampled every 1000th packet, using RouteViews data to map IPs to AS numbers. Relative volume is a measure of traffic towards the CDN, *not* of outbound traffic. Given the diversity of platforms, content, and customers served by the measured CDN, this is a reasonable measure of each network's *importance* as a source of client traffic. Relative volume is distinct from *load*, and instead indicates the relative popularity of a given network. These v values can be computed in a similar fashion for other services, *e.g.* incoming queries for a DNS resolver.

We attribute a portion of the relative volume that an AS contributes to the groups, g, that make up that AS. We define v_g in terms of the AS relative volume, v, the number of vantage points in the AS, P, and the number of vantage points in g, p_g, as:

$$v_g = v \cdot \frac{p_g}{P}.$$

Dividing by the count of vantage points assumes that the distribution of vantage points across groups within an AS matches the distribution of relative volume across sites. We make this assumption in the absence of visibility into the actual distribution of vantage points over the sub-groups.

Next, we define our weighted score $s_w(\Delta g)$ for each group as:

$$s_w(\Delta g) = s_{\text{raw}}(\Delta g) \cdot v_g. \quad (7)$$

These scores provide fine-grained, actionable data. Evaluating the scores across all groups allows operators to determine the impacts – both good and bad – of a configuration change.

Finally, for snapshots a and b, we define the net score, S:

$$S = \sum_{g \in G} s_w(\Delta g) \quad (8)$$

where G is the set of vantage point groups that is common to both snapshots. As discussed, the net score provides a coarse measure of the overall impact of the experimental configuration. If many groups improve we expect positive scores, and if many degrade, we expect negative scores. However, operationally, it may be more beneficial to inspect group-level score shifts, and make targeted configuration changes. We explore this approach in the next section.

4.2 DailyCatch Measurements

Here, we discuss the insights that can be provided by DailyCatch. DailyCatch is configured to test anycast blocks (as announced by a large, global CDN) that have been configured with the *transit* and *all-providers* configurations described in Section 3.4. Under the *transit* configuration, announcements are made to a restricted set of transit providers, while under the *all-providers* configuration, announcements are made to nearly all providers.

DailyCatch has been running in this way since August 2017. The case studies discussed here are from comparisons made during this period. The data is presented to demonstrate the utility of DailyCatch, rather than provide a comprehensive measurement of peering policies; as a result, the data is drawn from measurements taken throughout the period that DailyCatch has been in operation.

4.2.1 Biggest Movers. First, we examine the largest (both positive and negative) scores generated by DailyCatch for a single day during the test period. Table 2 shows the 3 largest and smallest scores, for both North American (left), and Europe (right); we mask AS numbers with letters to avoid revealing provider-level operational data. Broadly, we see that the groups with high scores tend to have higher v_g values, though not exclusively (Note Z / US-AZ). From this, we conclude that, in networks that matter most to the CDN, the experimental configuration (*i.e.*, announcing to more peers) improves performance. We can split the causes of the

[4] We evaluated the impact of using averages and other similar summary statistics and found no significant differences.

NA Group	Probes	RTT$_{tx}$	RTT$_{all}$	v_g	Score	EU Group	Probes	RTT$_{tx}$	RTT$_{all}$	v_g	Score
A / US-MN	5	27.8	11.4	.015	.008	D / FR	198	21.0	10.2	.025	.019
B / US-CO	13	28.5	10.9	.011	.006	E / GB	70	49.6	21.0	.024	.013
C / US-OR	4	19.9	12.2	.013	.005	F / IT	60	41.3	20.8	.010	.006
…											
B / US-MA	11	11.6	15.3	.009	−.003	X / ES	13	9.9	13.2	.011	−.003
Y / US-NC	2	26.8	145.3	.007	−.007	W / PT	11	16.9	40.8	.004	−.004
Z / US-AZ	13	21.8	27.8	.028	−.008	V / ES	7	7.7	37.4	.009	−.009

Table 2: Largest positive and negative scores for North America (NA) and Europe (EU) (from April 2018). Each row shows an individual group, where AS numbers are masked by letters.

scores shown in Table 2 into three broad categories: catchment shifts, path changes (but without catchment shifts), and transient, anomalous behavior.

As shown in Section 3, the addition of peering shifts catchments: upstream providers are likely to prefer to send traffic across a peering link. Groups B/US-CO, C/US-OR, D/FR, E/GB, and F/IT all see improved performance as a result of their catchments shifting to a closer site. However, groups B/US-MA, Y/US-NC, W/PT, and V/ES are all directed to farther away sites, and as a result, see worse performance. DailyCatch, by subdividing ASes by country or US state, allows us to identify that AS B sees changes in both directions; without this, the effect would likely be averaged out and hidden. Further, by weighting the changes by relative volume, DailyCatch identifies the sites where the addition of peering links would be most impactful. This targeted approach, enabled by group-level scoring and the collection of path information, is important to DailyCatch's practical, operational use.

We also identified changes in routing without a shift in catchment. Groups A/US-MN and Z/US-AZ see better or worse (respectively) performance resulting from taking faster or slower routes to the same site. Identifying path and catchment changes directly informs DailyCatch's need for traceroutes.

An artifact of our vantage point selection can be seen in the false positive visible in Group Y/US-NC. One of the two probes sees significantly worse performance. However, the other probe sees good behavior on an almost identical path. Group Y/US-NC's variation is a result of being a small group, where a measurement from one outlier can have a significant impact. Ultimately, we found that the benefit in visibility gained by including small groups outweighed the penalty of such outliers. We note, however, that the relatively low v_g weight for Y/US-NC has resulted in only a small negative contribution. The shifts seen in other small groups (*e.g.*, C/US-OR) show that small groups can provide meaningful data.

In this particular example, an operator may be satisfied with the improvements, but may consider altering the change to mitigate the poor performance for Z/US-AZ. By capturing a traceroute, DailyCatch allows the operator to identify the provider that AS Z's traffic arrived over. If, as in this example, AS Z is a direct peer, the announcement policy can be adjusted to not announce anycast to AS Z. This configuration can then be re-evaluated to ensure that there are no indirect impacts from excluding the provider. Other, moderate impacts (to B/US-MA, for example) may be deemed acceptable, due to the group's low relative volume.

In summary, many of the large changes shown in Table 2 result from significant shifts in both the *where* and *how* of client behavior. Many design decisions have also been validated: (i) subdividing by geolocation gives us insight into the behavior of a single network at multiple sites (*e.g.*, AS B); (ii) small probe groups give meaningful data (*e.g.*, AS C); and (iii) weighting by relative volume, v_g, surfaces relevant shifts. These variations show the complexity of anycast and provider interactions, emphasizing the importance of our sub-AS groupings.

4.2.2 Cause of Catchment Shifts. Another comparison surfaces from three notable positive score changes shown in Table 3. These three groups suffer from poor site selection when relying on transit providers, with all groups connecting to San Jose, despite a nearby site in Denver. Manual inspection shows that this is the result of a policy choice: the origin ASes prefer to route through a transit provider that is not available at the Denver site. The performance impact is clear: choosing a physically distant site inflates RTTs. In the *all-providers* configuration, probes now have access to a direct peering link, and select the better site.

This case study exposes the advantage of DailyCatch's approach. First, BGP collector data suggests that, in the *transit* configuration, these groups of clients should route to a site in Denver. By using active measurements, DailyCatch sees the impact of otherwise opaque BGP policies: it is not possible to determine *why* the clients do not take the available transit path, but it is sufficient to know that they don't. Finally, by announcing to additional peers, it is clear that performance can be improved directly, without explicit provider co-operation.

4.2.3 Mixed Impacts Scenarios. Figure 12 shows the difference in RTT in groups from three large North American ASes. The groups vary significantly in size, ranging from 4 probes to over 70, but we observe that group size doesn't correlate with observed performance. Impacts are not uniform within each AS, emphasizing the need for our grouping function in showing sub-AS-level variations. For example, in the leftmost AS, clients in Colorado achieved a nearly 13ms improvement, while clients in New Mexico had an almost 20ms penalty: a spread of more than 30ms. Similar behaviors are seen in the other ASes shown.

We note that groups that performed worse generally have a direct path that is performing worse. Ultimately, this is caused by providers *always* preferring peering links when available, which,

Group	Probes	RTT$_{tx}$	RTT$_{all}$	v_g	Score	Catchment$_{tx}$	Catchment$_{all}$
A / US-CO	21	35.36	10.70	0.0163	0.0098	San Jose	Denver
B / US-CO	4	45.03	20.23	0.0132	0.0066	San Jose	Denver
A / US-UT	3	35.82	21.00	0.0023	0.0009	San Jose	Denver

Table 3: Notable catchment shifts between *transit* and *all-peers* configurations.

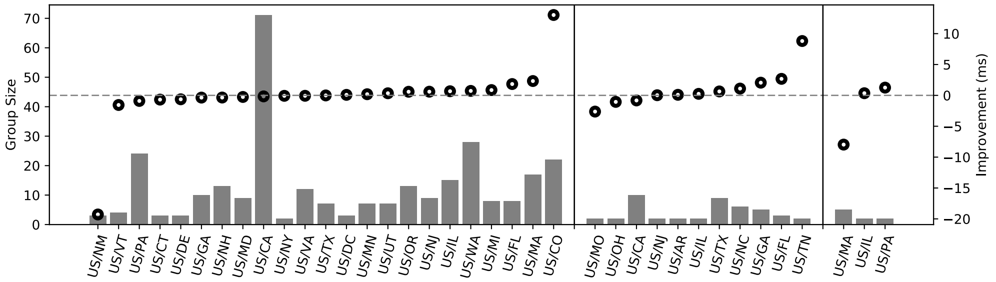


Figure 12: Change in median RTT for 3 large North American ASes. Each AS demonstrated mixed behavior across the groups. Gray bars indicate group size and circles indicate change in RTT.

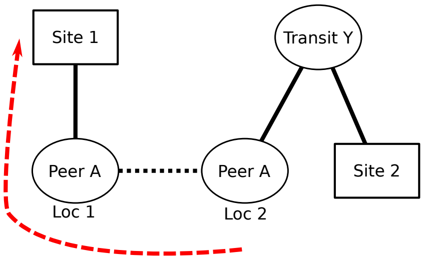


Figure 13: A large provider may prefer peering links that require traveling long distances over nearby transit, often altering catchment.

as illustrated in Figure 13, is a policy that may alter site catchments, and user-experienced latency, significantly. It is important to recognise the variations between sites: each has a different set of upstream providers and peers, depending on announcement configuration, business needs, and physical presence. In providing visibility into the behavior of individual groups, at specific sites, DailyCatch generates the data necessary to address many of the issues surfaced here, by adding peering, or reconfiguring announcements.

For the largest European networks, 3 saw significant improvements over 10ms, but none saw the mixed behavior seen in North America. This is potentially due to a different management approach, in which many European providers appear to operate separate ASes for each country or market. This difference is visible in the data: in Europe we observe a total of 731 ASes, spread over 75 countries. In contrast, in North America, we observe only 107 ASes, spread over 60 countries or US states. In Europe, 9 ASes appear in more than one country, while in North America, 19 appear in more than one country or US state. The average number of countries or US states per AS is 1.02 in Europe, compared with 1.71 in North America. One large provider appears in our data as no less than 6 distinct ASes. DailyCatch is agnostic to such differences, and is able to assess changes in behavior regardless of the particular network structure.

5 DISCUSSION

In this section, we discuss two important topics that relate to DailyCatch's use: how representative the RIPE Atlas platform is of the studied CDN, and the network misconfigurations that, when combined with the complex provider relationships discussed earlier, make an active measurement approach a requirement for the management of many-provider anycast announcement configurations.

5.1 RIPE Atlas Representation

At the time of writing, RIPE Atlas consists of over $10,000$ globally distributed probes. However, only 19% of IPv4 ASes seen by the studied CDN contain at least one RIPE Atlas probe. While this is greater than the raw AS coverage (5.958% of total ASes have at least 1 probe at time of writing [2]), when we consider the *weighted* fraction of requests from networks, we see even greater coverage: 61% of in-bound volume originates from networks with at least 1 probe.

However, this leaves a number of networks that have insufficient coverage. To explore these networks, we consider the *mismatch* between a network's popularity as seen by the CDN, and its probe coverage in RIPE Atlas. For a given AS, we define P as the number of probes it contains, and v to be its relative volume defined in Section 4.1. We then define the mismatch, M, as:

$$M = \frac{e^v}{1 + \frac{P}{P_{target}}}.$$

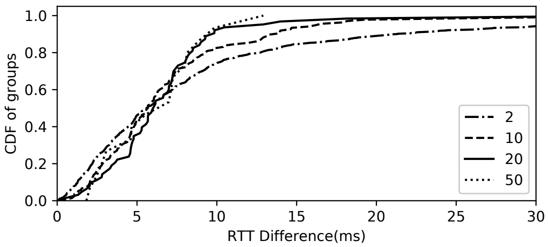


Figure 14: Changes in median RTT between the beacons and RIPE Atlas probes for various minimum group sizes

Rank	AS	v	P	M
1	G	0.0091	0	1.0211
2	H	0.0046	0	1.0107
3	I	0.0042	0	1.0097
9998	J	0.0299	326	0.0616
9999	K	0.0060	327	0.0584
10000	L	0.0041	349	0.0547

Table 4: The ASes with the largest and smallest rank mismatches.

Networks with at least P_{target} probes are considered well covered. To determine an appropriate value for P_{target}, we consider the error between our beacon data and RIPE Atlas measurements as in Section 3.3, but here we consider increasing the minimum number of RIPE Atlas probes required for each group.

Figure 14 shows the impact of minimum group size on the RTT error. As expected, increasing group size reduces error, with 20 probes providing comparable error to 50, with notable improvements over 10 probes, in particular above the 80th percentile. We therefore take our P_{target} to be 20.

The intuition of M is that significant networks covered by few or no probes will have high mismatch values, while networks with low relative volume (v) *or* high probe coverage (*i.e.*, $P > P_{target}$) result in a low M.

Table 4 presents the largest and smallest mismatch (M) values; we found these values to be consistent over a month, but we omit those findings due to space considerations. As expected, some networks that contribute a significant number of client requests are well covered by RIPE Atlas (*e.g.*, AS J and AS K are both significant end-user networks). The highly mismatched ASes are more interesting. Generally, these fall into three categories: cellular networks, networks in poorly covered regions, and networks that aren't sources of typical CDN traffic.

The top three highest mismatched ASes (AS G, H, and I) are large US cellular providers. While it is possible that these represent mixed networks [33], we confirm, by inspection of the CDN's HTTP access logs, that these are likely to be cellular clients. Low probe coverage in cellular networks is a known gap in the RIPE Atlas platform. The second category contains networks in geographic areas with known low probe coverage. This includes ASes from India and the Middle East. For both of these categories, the missing networks have significant relative volumes. However, there are relatively few of them: only 10 have v greater than .002. Such a small set of networks may be examined directly with other tools.

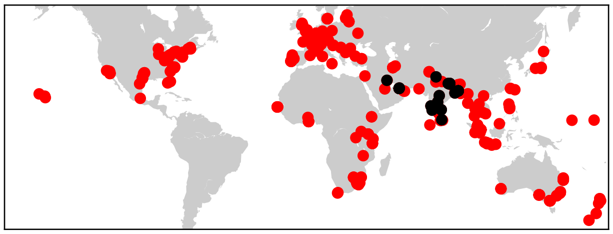


Figure 15: The RIPE Atlas catchment for a site in India before (black) and after (red) adding a new provider.

The final category of high-mismatch networks includes those that are not traditionally end-user networks for CDNs. These generally consisted of cloud-service providers in which users may host a vaiety of services. The majority of these networks had *no* probes at the time of measurement, but have notable relative volumes. Examination of requests from these networks suggests that they are primarily web services: for example, search engine crawlers, VPNs and web proxies, and generic cloud services. These are not representative of traditional CDN traffic and are out-of-scope for DailyCatch.

Despite some limitations, RIPE Atlas provides strong coverage for many important end-user networks, allowing DailyCatch to grant visibility into the provider-level impact of announcements. Ultimately, DailyCatch itself is generic, and could use any measurement platform that provides RTT measurements to arbitrary targets and catchment information for each vantage point.

5.2 Anycast Errors

Beyond the complex inter-domain behaviors described above, anycast networks must further contend with networks that behave in *unexpected* ways. We consider a real-world example involving the global anycast announcements from the studied CDN. Figure 15 maps the catchment of a site in India before (black dots) and after (red dots) announcing to a single new peer. In this case, this change shifted the catchment, and resulted in nearly all traffic entering the network from a single interconnect for the provider, likely due to an upstream misconfiguration which liberally re-announced the block.

Figure 16 further examines the performance impacts from this configuration change, showing the percentage increase in RTTs between these configurations. Of the 54 impacted ASes, we can see that for about 20% of them saw little degradation, and indeed some did not change catchment. However, beyond that, we notice significant degradation, up to about 1,200% (13ms to over 170ms), coming from an AS that was previously served in the United States that now traveled to India.

The impact of any announcement change is tangled: an upstream provider's incentives are often not the same as the anycast operator's, and network misconfigurations are often opaque, in particular

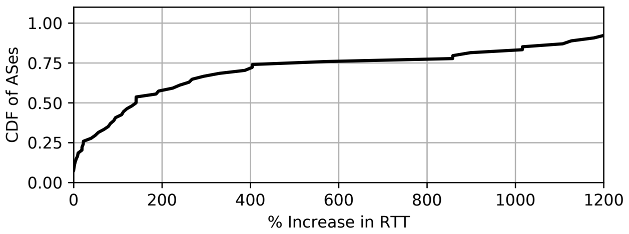


Figure 16: The % increase in RTT by AS for those impacted by the misconfiguration.

when there are many providers. Inferring the behavior of all networks, on a global scale, requires significant modelling and measurement effort, alongside prior knowledge of the likely routing policies of third-party networks. Routing tables are constantly changing [1], limiting the use of any particular heuristics. DailyCatch provides visibility, revealing the source of such issues: this makes it an effective tool for their detection and mitigation.

6 RELATED WORK

Anycast has been a significant field of study within the networking community. Many studies have focused on the behavior of anycast for DNS infrastructure [14, 17, 29–31]. Other works have studied non-DNS anycast deployments [11, 18, 23], or systems built on top of anycast [6–9, 21]. In [36], the authors examine anycast stability, and how it impacts file download completion. These works are largely orthogonal: with the visibility that DailyCatch provides, many of these systems could likely be improved.

In [10], the authors explore the implications of anycast configurations and attempt to provide a guiding heuristic for future deployments, ultimately suggesting that a single provider would provide consistent behavior. In [15], the authors determine the number of anycast sites necessary to achieve reasonable latency, characterising the diminishing returns of new sites, showing, via unicast measurements, that anycast may make poor decisions. In [11, 12] the authors propose a DNS based work around for poorly performing anycast. More recently, [27] examined anycast performance and proposed a BGP-communities based solution for controlling it.

While important for understanding the nature of the choices made when using anycast, many of these systems rely on comparison against unicast behavior, which may not be achievable without significant infrastructure [12] or not yet deployed features [27]. We argue, instead, for an approach that pursues behavior that can be achieved today, via explicit anycast configurations.

In [16] the authors present Verfploeter, a system for mapping anycast catchments using ping responses directed to anycast addresses. While an extremely powerful tool for mapping catchments, Verfploeter does not provide RTT or *path* information, both critical components in managing anycast performance.

Systems such as Google's Espresso [38] and Facebook's Edge-Fabric [34] send small samples of traffic on alternative paths to provide regular feedback. However, they are designed for managing *egress* traffic. Here, we are considering *ingress* traffic, which relies on changes in BGP announcement configuration.

There has been a significant body of work exploring the economic and performance trade-offs of Internet connectivity [5, 37]. Others have examined the increasing flatness of the Internet topology [13, 20, 22, 26, 35]. In DailyCatch, we are only concerned anycast performance impacts: different providers behave differently. Measuring and responding to conditions as providers are added – regardless of any underlying trends – is essential.

WhyHigh [25] is a system that helps to find problematic behaviors by prefix, surfacing the most troublesome, and helping to determine their root cause. While similar, DailyCatch focuses on anycast networks, and in particular on controlled experiments to compare anycast configurations. In [12], the authors present a generalized CDN measurement system. DailyCatch is orthogonal, as it focuses on a method for tuning anycast policy, an issue [12] agrees is complex.

7 CONCLUSION

In this paper, we have presented a measurement and optimization proposal for large, many-provider anycast networks, in the context of a global CDN. We have shown that anycast networks with more network providers interact with the Internet in an observably different way than those with fewer providers, with incoming traffic using a more diverse set of, more often short, paths. We further demonstrated how announcement configurations can be manipulated to provide performance improvements, and observed that care must be taken to avoid suffering performance degradation.

Finally, we presented DailyCatch, a tool for performing active measurements of anycast configurations, which provides sub-AS-level visibility into the impacts of changes. We demonstrated that DailyCatch is able to detect both performance improvements, as well as degradations, that may arise from idiosyncratic network policies. Finally, we provided thorough measurement of the coverage of the RIPE Atlas platform, providing clarity into which networks are well represented.

REFERENCES

[1] CIDR report. https://www.cidr-report.org/as2.0/.
[2] RIPE Atlas. https://atlas.ripe.net.
[3] root-servers.org. https://root-servers.org/.
[4] RouteViews. http://www.routeviews.org.
[5] A. Ahmed, Z. Shafiq, H. Bedi, and A. R. Khakpour. Peering vs. transit: Performance comparison of peering and transit interconnections. *Proc. of ICNP '17*, 2017.
[6] Z. Al-Qudah, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe. Anycast-aware transport for content delivery networks. In *Proc. of WWW '09*, 2009.
[7] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van der Merwe. Anycast CDNs revisited. In *Proc. of WWW '08*, 2008.
[8] H. A. Alzoubi, S. Lee, M. Rabinovich, O. Spatscheck, and J. Van Der Merwe. A practical architecture for an anycast CDN. *ACM Trans. Web*, 2011.
[9] H. Ballani and P. Francis. Towards a global IP anycast service. *SIGCOMM CCR*, 2005.
[10] H. Ballani, P. Francis, and S. Ratnasamy. A measurement-based deployment proposal for IP anycast. In *Proc. of IMC '06*, 2006.
[11] M. Calder, A. Flavel, E. Katz-Bassett, R. Mahajan, and J. Padhye. Analyzing the performance of an anycast CDN. In *Proc. of IMC '15*, 2015.
[12] M. Calder, R. Gao, M. Schröder, R. Stewart, J. Padhye, R. Mahajan, G. Ananthanarayanan, and E. Katz-Bassett. Odin: Microsoft's scalable fault-tolerant CDN measurement system. In *Proc. of NSDI '18*, 2018.
[13] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan. Are we one hop away from a better Internet? In *Proc. of IMC '15*, 2015.
[14] L. Colitti, E. Romijn, H. Uijterwaal, and A. Robachevsky. Evaluating the effects of anycast on DNS root name servers. In *RIPE-393*, 2006.
[15] R. de Oliveira Schmidt, J. Heidemann, and J. H. Kuipers. Anycast latency: How many sites are enough? In *Proc. of PAM '17*, pages 188–200. Springer, 2017.

[16] W. B. de Vries, R. de O. Schmidt, W. Hardaker, J. Heidemann, P.-T. de Boer, and A. Pras. Broad and load-aware anycast mapping with Verfploeter. In *Proc. of IMC '17*, 2017.

[17] X. Fan, JHeidemann, and R. Govindan. Evaluating anycast in the domain name system. In *Proc. of INFOCOM '13*, 2013.

[18] A. Flavel, P. Mani, D. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev. FastRoute: A scalable load-aware anycast routing architecture for modern CDNs. In *Proc. of NSDI '15*, 2015.

[19] R. Fontugne, A. Shah, and E. Aben. The (thin) bridges of as connectivity: Measuring dependency using as hegemony. In *Proc. of PAM '18*, pages 216–227. Springer, 2018.

[20] R. Fontugne, A. Shah, and E. Aben. The (thin) bridges of as connectivity: Measuring dependency using as hegemony. In *Proc. of PAM '18*, 2018.

[21] M. J. Freedman, K. Lakshminarayanan, and D. Mazières. OASIS: Anycast for any service. In *Proc. of NSDI '06*, 2006.

[22] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. The flattening Internet topology: Natural evolution, unsightly barnacles or contrived collapse? In *Proc. of PAM '08*, 2008.

[23] D. Giordano, D. Cicalese, A. Finamore, M. Mellia, M. M. Munafò, D. Z. Joumblatt, and D. Rossi. A first characterization of anycast traffic from passive traces. In *Proc. of TMA 2016*, 2016.

[24] T. Holterbach, E. Aben, C. Pelsser, R. Bush, and L. Vanbever. Measurement Vantage Point Selection Using A Similarity Metric. In *Proc. of ANRW '17*. ACM, 2017.

[25] R. Krishnan, H. V. Madhyastha, S. Jain, S. Srinivasan, A. Krishnamurthy, T. Anderson, and J. Gao. Moving beyond end-to-end path information to optimize CDN performance. In *Proc. of IMC '09*, 2009.

[26] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In *Proc. of SIGCOMM '10*, 2010.

[27] Z. Li, D. Levin, N. Spring, and B. Bhattacharjee. Internet anycast: Performance, problems, & potential. In *Proc. of SIGCOMM '18*, 2018.

[28] K. Lindqvist and J. Abley. Operation of anycast services. RFC 4786, Dec. 2006.

[29] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy. Two days in the life of the DNS anycast root servers. In *Proc. of PAM '07*, 2007.

[30] G. Moura, R. d. O. Schmidt, J. Heidemann, W. B. de Vries, M. Muller, L. Wei, and C. Hesselman. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In *Proc. of IMC '16*.

[31] M. Müller, G. C. M. Moura, R. de O. Schmidt, and J. Heidemann. Recursives in the wild: Engineering authoritative DNS servers. In *Proc. of IMC '17*, 2017.

[32] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. In search of the elusive ground truth: the internet's as-level connectivity structure. In *ACM SIGMETRICS Performance Evaluation Review*, volume 36, pages 217–228. ACM, 2008.

[33] J. P. Rula, F. E. Bustamante, and M. Steiner. Cell spotting: Studying the role of cellular networks in the Internet. In *Proc. of IMC '17*, 2017.

[34] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng. Engineering egress with edge fabric: Steering oceans of content to the world. In *Proc. of SIGCOMM '17*, 2017.

[35] Y. Shavitt and U. Weinsberg. Topological trends of Internet content providers. In *Proc. of SIMPLEX '12*, 2012.

[36] L. Wei and J. Heidemann. Does anycast hang up on you? In *Proc. of TMA '17*, 2017.

[37] F. Wohlfart, N. Chatzis, C. Dabanoglu, G. Carle, and W. Willinger. Leveraging interconnections for performance: The serving infrastructure of a large CDN. In *Proc. of SIGCOMM '18*, 2018.

[38] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat. Taking the edge off with Espresso: Scale, reliability and programmability for global Internet peering. In *Proc. of SIGCOMM '17*, 2017.