Bidirectional Anycast/Unicast Probing (BAUP): Optimizing CDN Anycast

Lan Wei* Marcel Flores† Harkeerat Bedi† John Heidemann*
* USC/Information Sciences Institute
{weilan, johnh}@isi.edu
† Verizon Digital Media Services
{marcel.flores, harkeerat.bedi}@verizondigitalmedia.com

***Abstract*—IP anycast is widely used today in Content Delivery Networks (CDNs) and for Domain Name System (DNS) to provide efficient service to clients from multiple physical points-of-presence (PoPs). Anycast depends on BGP routing to map users to PoPs, so anycast efficiency depends on both the CDN operator and the routing policies of other ISPs. Detecting and diagnosing inefficiency is challenging in this distributed environment. We propose *Bidirectional Anycast/Unicast Probing (BAUP)*, a new approach that detects anycast routing problems by comparing anycast and unicast latencies. BAUP measures latency to help us identify problems experienced by clients, triggering traceroutes to localize the cause and suggest opportunities for improvement. Evaluating BAUP on a large, commercial CDN, we show that problems happens to 1.59% of observers, and we find multiple opportunities to improve service. Prompted by our work, the CDN changed peering policy and was able to significantly reduce latency, cutting median latency in half (40 ms to 16 ms) for regions with more than 100k users.**

I. Introduction

Content-Delivery Networks (CDNs) and Domain Name System (DNS) operators use globally distributed PoPs (Points of Presence) to bring content closer to users. Ideally, users are directed to the PoP that can provide the lowest possible latency for the desired content. Many CDNs [11], [15], [9], [18] and DNS services (such as the DNS root [29]) use *IP anycast* to direct users to PoPs. With anycast, services are announced on one IP address (or block of addresses), and Internet routing associates users to PoPs by BGP. Border Gateway Protocol (BGP) is influenced by routing policies set by the CDN and ISPs [7]. Previous evaluations of CDNs suggest that anycast does not always find the lowest latency [22], [9], and studies of anycast infrastructure suggest that BGP does not always select lowest latency [19], [24], [31], [23].

Operators of CDNs and DNS services need to identify and correct latency problems in the network. While multiple large CDNs directly serving 1k to 2k ASes [1], [8], [10], and there are more than 1000 root DNS instances [29], with more than 67k ASes on the Internet [3], the majority are served indirectly through other ASes. In addition, some CDNs and DNS providers operate fewer, larger PoPs, or cannot deploy in some ASes or countries due to financial or legal constraints, so optimizing performance across multiple ASes is essential.

It is challenging to detect problems in IP anycast deployments, much less identify root causes and deploy corrections. Problem identification is difficult because one must distinguish between large latencies that are due to problems (say, a path that misses a shorter route) from latency that is inherent (for example, users connecting via satellite or over long distances). Root causes are challenging to find because even though measurement can provide latencies and paths, we do not know *why* problems occur. Finally, once problems have been identified (for example, a provider known to have frequent congestion), the CDN must determine solutions to those problems. Solutions are not always possible and can be difficult to determine, particularly when the problem is multiple hops away from the CDN.

Our first contribution is to design *Bidirectional Anycast/Unicast Probing, BAUP*, a method to evaluate anycast performance for CDNs and DNS from both inside and outside. BAUP allows operators to detect potential performance problems caused by congestion or unnecessary routing detour and learn an optional better route. BAUP first *detects* potential problems from differences in anycast and unicast latency (§IV). When a potential problem is detected, it then classifies the problem with traceroutes, by checking both the forward and reverse paths for both anycast and unicast, between vantage points and the CDN, while considering potential path asymmetry [32]. We show that this information allows us to identify slow hops and circuitous paths, two classes of problems that occur in anycast CDNs (§V-A).

Our second contribution is to *evaluate how often performance problems occur* for a large, commercial CDN (§V). We see that about 1.59% of observers show potential latency problems. While this number seems small, the CDN implemented changes in response to our work and saw improvements across 91 ASes in 19 countries, affecting more than 100k users.

Our final contribution is show that BAUP can result in noticable improvements to service. BAUP is a tool to help CDN and DNS operators detect and correct congestion and unnecessary routing detours in complicated and changing global routing, ultimately improving tail-latency [13] for their users. We find three such cases where solutions are possible in our candidate CDN to improve performance. While the constraints of operational networks mean that routing changes cannot always be made, our work prompted one set of peering policy changes in CDN deployment (§VI). After this change, *latency was significantly reduced in regions with a large number of users*, falling by half, from 40 ms to 16 ms. This large improvement was in tail latency—before the the improvement, our observers in the 100k users that improved showed median

978-3-903176-27-0 ©2020 IFIP

latency at 86%ile of all users.

The measurements in this paper use RIPE Atlas. While we cannot make CDN-side data public, all external measurements towards the CDN is publicly available [28]. To preserve privacy, IP addresses in this paper use prefix-preserving anonymization [16], [2], and we replace Autonomous System (AS) numbers with letters.

II. Problem Statement

Two common problems in anycast CDNs are paths that use *congested links*, and *high-latency paths* that take more hops (or higher-latency hops) than necessary. We call using such a path as *unnecessary routing detour*. In both cases, end-users experience reduced performance. We can detect both of these problems with BAUP.

Congested links occur when a path has persistent congestion, so traffic suffers queueing delay. Such congestion can often occur at internal links, private peerings, or IXPs [14] that have insufficient capacity.

High-latency paths occur when the selected path has larger latency than other possible paths for reasons other than congestion: typically because it follows more hops or larger latency hops. Anycast paths are selected at the mercy of the BGP, and while BGP selects to minimize hop counts, it does not always provide the lowest latency, and routing policy can override minimal-hop-count paths.

Other problems, like high-loss links, are outside our scope.

A. Observations to Find Problems

We define congested links and high-latency paths as problems of interest, but they cannot be directly observed. We next define two more specific behaviors we can actually measure with network traceroutes: a *slow hop* and a *circuitous path*.

A *slow hop* is a hop in the traceroute which show unusually high latency—the specific threshold for abnormal is a function of the path, described in §IV-C1. Link congestion, long physical distance, or high-latency in the return path can lead to a slow hop observed in a traceroute. Our measurements search for slow-hops that can be fixed, and try to identify and dismiss distance-based latency that cannot be improved. We classify slow hops by where they occur: *intra-AS and inter-AS slow hops* happen inside an AS and between ASes, respectively. *Near-CDN slow hops* are a special case of inter-AS slow hops where the CDN operator can change peering policies directly.

A *circuitous path* is a high-latency path that occurs and we know a lower-latency path exists. In our observation, a circuitous path contains different hops from the alternative path, and it has longer end-to-end latency measured.

Figure 1 shows hops and circuitous paths by looking for *asymmetric AU latency* and by checking both the forward and reverse paths (considering the Internet asymmetry) with both anycast and unicast. We describe our detection methods in §IV and show examples of slow hops and circuitous paths in §V-A. Our goal is to find problems that a CDN can address (§IV-D), that is, slow hops or circuitous paths where other routes exist. We call these cases *improvable latency*.

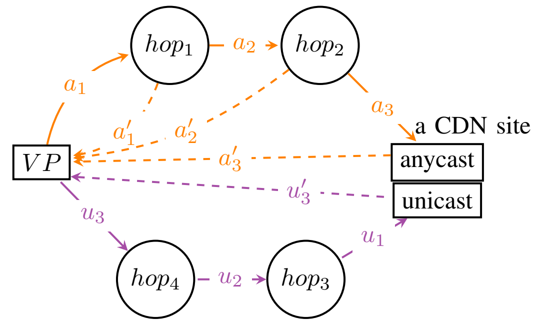

Fig. 1: Four one-way delays in BAUP traceroute, including VP to anycast CDN, anycast CDN to VP, VP to unicast CDN, unicast CDN to VP. All four can be **different** from each other.

III. RTT Inequality between Anycast/Unicast

To provide context for how our new probing method detects problems, we next explore *why would anycast and unicast addresses ever produce unequal RTTs from the same clients?* This question can be covered by a larger question: How can the round-trip time be different towards one location, if the sender measures twice towards two different IP addresses of this location? The fact is the path taken to connect to two different addresses from different BGP prefixes can be different, no matter the physical location of the destination. Therefore the round-trip time taken can also be different. Next, we examine in detail why the route can be different, as it is determined by two major factors—BGP, and network asymmetry.

In particular, we note that in a single round-trip, there are two constituent one-way trips. So if measuring from a Vantage Point (VP) to a CDN, one can target two different IP addresses at the CDN, the anycast and unicast addresses. Together, there are two potentially different round-trips, and four one-way trips. Using Figure 1 as an example, there are four one-way trips in the graph, VP to anycast CDN (via hop_1 and hop_2), anycast CDN to VP (by $a_3\prime$ with no hops marked in graph), VP to unicast CDN (via hop_4 and hop_3), unicast CDN to VP (by u'_3 with no hops marked in graph).

BGP determines the route towards the unicast and anycast addresses, and the route can be different for each addresses. In Figure 1, when the VP connects to CDN site via its anycast address, the forwarding path will route to hop_1 first. BGP selects this hop_1 based on factors such as AS path length and local preference, which in turn may be determine by the originating announcements and subsequent propogation. The same goes when the VP try reaching the same CDN site but via its unicast address with the first hop as hop_4. Factors such as AS path length and local preference can vary based on the destination address. This difference in address may result in hop_1 and hop_4 being different. For the same reason, hop_2 and hop_3 may vary as well. In fact, the count of hops may also differ in the two forward paths to the anycast and unicast address of one CDN site. So now we know, the forw portion of the two round-trips may be different.

With asymmetric network routing, the reverse path may differ from the forward path [12]. Although the two forward paths being different sufficiently proves the two round-trips to unicast and anycast can be different. Since the two forward paths may be different, and the reverse different from the forward, all four single one-way trips (VP to anycast CDN, anycast CDN to VP, VP to unicast CDN, unicast CDN to VP) may be different from each other.

IV. Bidirectional Anycast/Unicast Probing

Bidirectional Anycast/Unicast Probing (BAUP) is a new method to observe slow hops and circuitous routes, suggesting congested links and high-latency paths that perhaps can be avoided. We use *Vantage Points* (VPs) that carry out active latency measurements to anycast and unicast addresses in the CDN, providing two latency estimates. We *detect* potential routing problems when a VP sees consistently higher latency on one of those two paths. Once a potential problem has been detected, we take bidirectional traceroutes (three in total), including VP to the unicast CDN, unicast CDN to VP, VP to the anycast CDN. This information helps us identify problems and suggest potential changes to routing that can allow the CDN to improve performance.

A. BAUP Measurements

BAUP requires VPs that can carry out active measurements (pings and traceroutes) under our control. We assume some control at the CDN: we assume each CDN PoP has a unique unicast address in addition to its shared anycast address, and that the CDN can send traceroutes out of the unicast address. We use VPs from RIPE Atlas (where they are called "probes") to set up the BAUP, and we work with a commercial CDN network. Our study maximizes the path set between users and CDN by using all available VPs that does not have duplicate source IP addresses.

We first identify each VP's catchment in the CDN's anycast network. Methods for such identification may vary by CDN—DNS services may use NSID queries [4], [17], or one may use a tool like Verfploeter [12]. We determine VP catchment by taking traceroutes from our VPs to the CDN anycast address and searching a database of CDN BGP sessions for the final hop in the traceroute that lies outside of the CDN network. If we find a match, we label the VP as within the catchment of the unique PoP where the BGP session exists.

We initially take latency measurements (pings) from the VP to both the anycast and unicast addresses for the VP's cachement. We use differences in that latency to detect potential improvements, as described next in §IV-B.

For VPs that show potential improvement, we traceroute from the VP to the unicast and anycast addresses in the CDN, and from the CDN's unicast address to the VP. We would like to traceroute from the CDN's anycast address, but because the anycast addresses are in production, we cannot easily take non-operational measurements there. We study the path with information of both IP addresses and ASNs suggested by Route Views [30]. IP addresses and ASNs shown in the paper are encrypted in prefix-preserving method [16], [2].

BAUP thus provides us hop-wise information about three one-way paths, as shown in Figure 1: VP to the CDN's anycast address, VP to its cachement's unicast address in the CDN, and from that unicast CDN address to the VP.

B. Detecting Improvable Latency

BAUP detects improvable latency by finding asymmetry between the A- and U-latency. We consider U-latency smaller than A-latency the indicator of an improvable latency, since CDN users reach CDN in A-probing.

We can define several types of latency from our measurements (Figure 1). *A-probing*, the VP-to-anycast latency, defines RTT_A from $a + a'$ in Figure 1, where a is the end-to-end unidirectional latency from VP to the CDN, and a' is the unidirectional latency on the reverse path from the CDN to the VP. *U-probing* gives the VP-unicast latency, with RTT_U from $u+u'$, with u and u' the VP-to-unicast CDN and unicast CDN-to-VP latencies, respectively. We detect improvable routing when $RTT_U < RTT_A$.

As individual pings are often noisy, we repeat A- and U-probing to look for *consistently unequal RTTs*. We define large enough as ($\delta > 10ms$) or ($\delta > 0.15 * max(RTT_A, RTT_U)$ and $\delta > 5ms$). We chose these two factors, absolute gain of 10 ms and proportional gain of $0.15 \times RTT$, to focus on improvements that are meaningful to users and therefore worth attention. We define consistent results when 80% of observations meet this criteria. The specific thresholds are not critical: 10 ms, 0.15, and 80% are based on operational experience at balancing true and false positives, and others may choose a different threshold. In our experiments we observe RTT every two hours for 48 hours, giving 24 samples, but we think 12 samples and 24 hours is sufficient—the requirement is to observe long enough to identify network topology and not just transient congestion.

C. Locating the Problems

After we detect VPs with the potential for latency improvement (§IV-B), we next need to localize the problem, identifying a specific slow hop or circuitous route. Our three traceroutes (Figure 1) provide information to identify these events. We first look for slow hops, and if we find none, it suggests a circuitous path (longer networking distance without specific slow hops). We next review how we find these events, using examples we expand upon later in §V-A.

1) Detecting Slow Hops

We find slow hops by examining traceroutes. Traceroutes report the IP address of each hop, and the RTT from the source to that hop. *Slow hops* occur when there is a sudden increase in latency (for example, the bold hop-to-hop marked in Table II, Table III, Table IV). For each traceroute record, we compute the incremental RTT change (usually a small rise) hop by hop. If for a specific hop, its incremental change from its previous hop is larger than the median plus twice the median absolute deviation of all incremental RTT change in a traceroute record, we consider this hop as a slow hop.

The observation of a slow hop can point back to three possible root-causes, a distant next hop, a congested link, or a high-latency reverse path. Of these, a distant next hop is not a problem, but perhaps unavoidable to bridge the distance between source and destination. (On the other hand, a shorter U-path will prove that the slow-hop can be avoided and is therefore not due to physical distance.). However, a congested link or high-latency path are problems that can perhaps be addressed by taking different paths. However, in some cases, subsequent hops of a slow hop may show *lower* RTTs than this slow hop, suggesting a long delay in the reverse path of this slow hop (and this reverse path is not shared by the subsequent hops). We consider such cases *false slow hops*, and discuss how we identify and avoid false slow hops below.

2) How RTT Surge Reveals A Slow Hop

To show the forming of a slow hop, in Figure 1, we have two RTTs between (VP, hop1) and between (VP, hop2), and we name them R_1 (a_1+a'_1) and R_2 (a_1+a_2+a'_2). If we assume R_1 is a reasonable value and R_2 is surprisingly larger than R_1, this could mean either a_2 is large or ($a'_2 - a'_1$) is large. A large a_2 means either hop_1 is congested or path of a_2 is long, but we rule out hop_1 being congested because R_1 is an assumed reasonable value. A large ($a'_2 - a'_1$) means that either hop_2 is congested, or a_2 takes an inferior route, and a_1 does not take an inferior route as assumed. Symbols such as a_n and u_n in Figure 1 indicate the path segments where long latency can occur. We do not need the latency of the exact path segment to detect improvable latency.

3) Avoiding False Slow-Hops

Some hops appear "slow", but do not affect the end-to-end RTT. The reverse paths can be different for different traceroute hops, and the reverse path from an intermediate router may not overlap the reverse path from the CDN. A false slow-hop will occur if it has a high latency reverse path that does not overlap with the destination's reverse path. We exclude those false slow-hops because their increased latency does not pass to later hops. Fortunately, true slow-hops (due to congestion, distance, or other consistent latency) can be determined because their latency appers in subsequent hops.

In Figure 1, we consider 3 RTTs between (VP, hop1) and between (VP, hop2), and (VP, CDN-anycast) and we name them R_1 (a_1+a'_1), R_2 (a_1+a_2+a'_2), R_3 (a_1+a_2+a_3+a'_3). If we assume R_2 is surprisingly larger than R_1, making hop_2 look like a slow hop. But R_3 is much smaller than R_2. We learn that a'_3 is smaller than a'_2, which means CDN takes a much faster return route than the hops before. A slow hop like this which does not affect the final VP-CDN RTT, we call it a false slow hop. Table V provides a specific example of false slow hop, we note the hop marked with a strikethrough: although RTT of this hop increases by about 10ms from its previous hop, this increase does not continue to its next hop (with 16.54ms dropping to 10.23ms, 17.0ms to 13.18ms). This suggests the sudden RTT increase of this hop is potentially due to this hop taking a slower reverse path not used by the final destination.

4) Circuitous Path Detection

A VP suffers from a circuitous path when no slow hops are detected but there is still improvable latency. If we look at the middle section of Table V, we can see the VP-to-unicast has a 9ms RTT, much shorter than this hop's 17ms. In this case, packets on the anycast CDN to VP (not included in BAUP) path encounter additional delay. Although we cannot see the details of the path from anycast-CDN-to-VP, the fact that VP-to-anycast and VP-to-unicast match (left section and middle section being the same), means the unicast-CDN-to-VP does not encounter the same latency as the anycast-CDN-to-VP.

IP aliasing can result in inaccurate ASes in traceroutes [21]. Although there has been progress reducing aliasing [27], it seems impossible to eliminate. Fortunately, our work keeps IP addresses to study the path, and requires only rough matches of ASes and /24 prefixes to classify problems (§V-A), and does not require correct AS identification for mitigations (§IV-D).

D. From Problems to Solutions

A CDN can resolve slow hops and circuitous routes by changing its routing policies, or asking its peers to chage. The presence of an existing, lower-latency U-path suggests a better path does exist.

For circuitous paths, when U-probing suggest a better route, the CDN can change its anycast traffic to follow the path shown in U-probing to reduce anycast latency.

For paths with slow-hops, the CDN needs to influence routing to avoid the slow hop, perhaps by not announcing at a given PoP or to a given provider at that PoP, by poisoning the route, or by prepending at that PoP. Again, the existence of the lower-latency U-path motivates change by proving a better path exists, but U-path may not be the only solution path for operators to follow. As long as the slow hops are avoided, the operators may find other good paths rather than U-path. The best mitigation varies depending on the location of the problem: if the slow hop is in a network that directly peers with the CDN, it has immediate control over use of that network. It is more difficult to make policy changes that route around slow hops that are multiple hops from the CDN.

In wide-area routing, BAUP must be prepared to handle load balancing in the WAN and at anycast sites and potential routing changes. Prior work has shown that catchment changes are rare [33], so wide-area, load-balanced links and routing changes are unlikely to interfere with BAUP analysis. Load balancing inside an anycast site is common, but unlikely to offer alternate paths that woud appear in BAUP's wide-area analysis. BAUP can detect and ignore cases where the A- and U-paths end at different sites.

We show specific case studies next (§V-A) and later show an example where a CDN was able to provide a significant reduction in latency to certain regions (§VI). In fact, we show that improvements to detect the problems we found actually benefited a broader set of clients, not all directly detected by BAUP. The advantage of BAUP is to find VPs that have opportunities for lower latency paths. It serves to automate

problem	CDN PoP	RTT_A	RTT_U
intra-AS	FRA	32.91	27.32
inter-AS	FRA	20.33	11.38
near-CDN	VIE	25.86	2.20
circuitous path	FRA	22.47	9.44

TABLE I: Basic information about routing problems. PoP are given as near-by airport codes.

identification of such locations, allowing CDN operators to focus on networks that are likely to show improvement.

Limitations: Our methodology has two limitations. First, BAUP cannot discover all available path between a single VP and the CDN. Instead it knows only the current A-path and the alternate U-path. Future work may study one-way latency (BAUP studies round-trip) to isolate each direction to find more improvable cases and use other methods to find more alternate paths. Second, sometimes it may be hard for the operator to use the U-path for anycast. The majority improvements we found were for slow hops. For slow-hops, the operator does not need to adopt the U-path (and sometimes cannot, perhaps if load balancers hash A- and U-paths differently). In these cases, the operator must influence the A-path to improve the latency.

V. Results

We next evaluate a CDN from all available RIPE Atlas probes (our VPs); our goal is to identify opportunities to reduce latency. Measurements begin at 2019-07-29T00:00 UTC and run for 48 hours, with each VP running an AU Probe every two hours (so 24 observations per VP). We confirm the anycast catchment and valid RTT_A and RTT_U values for comparison (see §IV-B) for 8350 of the 9566 probes.

Given the concentration of RIPE Atlas VPs in Europe [5], it is more likely that we will find problems there. The goal of our experiments is to show BAUP finds real-world problems and to provide a lower-bound into how many problems exist. We claim only a lower bound on the number of anycast problems we find, not tight global estimate, so any European emphasis in our data does not change our conclusions.

A. Case Studies: Using BAUP to Identify Problems

Before our general results, we show examples problems (from §II) BAUP revealed. Table I provides example latencies.

1) Intra-AS Slow Hop

Our first example is an intra-AS slow hop at Table I, visible as a 5 ms difference in RTT_A vs. RTT_U (background colors show different ASes). Table II shows each hop of paths in and out of the CDN. All three paths have unreported hops (the dashes), and the slow hop is in AS-A (a large Transit provider) on the inbound path.

This problem may be inside AS-A , or its reverse path (see §IV-C). Since the CDN peers directly with AS-A , the CDN may be able to influence the path.

2) Inter-AS Slow Hop

Slow hops may also happen between ASes, not only inside one AS. Our second example is an inter-AS slow hop with a 9 ms difference in RTT_A and RTT_U (Table I). Table III shows a slow hop when a packet leaves AS-D and enters AS-E (a large transit provider).

Although this problem may be at AS-E or its reverse path, U-probing shows a much faster path through AS-F (a route through an Internet exchange provider). While the CDN does not currently announce anycast to this provider, they may consider adding them to take advantage of this direct route.

3) Problem near the CDN hop

A case we are especially interested in are slow hops near the CDN, since they can often be addressed easily. Our third example is a slow hop identified with a 23 ms difference in RTT_A and RTT_U (Table I). In Table IV, we cannot tell the specific location of the slow hop, it may be the fifth hop of AS-H or the fourth hop, not shown in the traceroute information.

Luckily, since the slow hop is near CDN, U-probing suggests an alternative path through a different provider that will reduce the 25ms latency to 2ms.

4) A Circuitous Path

Next, we look at an example of circuitous path. We identify this problem because of a 13 ms difference in RTT_A and RTT_U (Table I). We can infer the existence of a circuitous path in the anycast-CDN-to-VP, although our data does not provide the details of the path. BAUP provides three of the four possible one-way delays, but we lack *anycast CDN to VP*. Table V shows the VP to anycast CDN (a) and VP to unicast CDN (u) are the same path, but round-trip-time of former is much larger. We also learn the path unicast CDN to VP ($u\prime$). We therefore infer that the higher latency occurs in the one-way delay from anycast CDN to VP ($a\prime$). To express what we learn mathematically, if (a + $a\prime$) $>$ (u + $u\prime$) and $a = u$, then it must be that $a\prime > u\prime$.

U-probing tells us that unicast-CDN-to-VP is faster than the anycast-CDN-to-VP. It proposes an explicit path suggestion via AS-F and AS-G , suggesting that the CDN operator use a path via AS-F to improve performance.

B. How Often Does BAUP Find Latency Differences?

Generally BGP works well to select anycast PoPs, with AU detecting unequal latency relatively infrequently. Table VI show the results of our evaluation: most of the time (more than 95%), the A- and U-paths show similar latencies. About 4.6% show differences and therefore potential routing problems. When RTTs are unequal, A-probing is faster than U-probing about twice as often, suggesting that anycast routing is already generally well optimized.

C. Root Causes and Mitigations

We next examine the 133 cases where there is a difference and U-Probing is faster, since those are cases where anycast

Vantage Points											
src	**AS**	**IP**	**RTT**	**src**	**AS**	**IP**	**RTT**	**dst**	**AS**	**IP**	**RTT**
↓	—	207.213.128.248	1.43	↓	—	207.213.128.248	1.32	↑			
↓	AS-B	115.66.46.99	1.38	↓	AS-B	115.66.46.99	3.3	↑	—	—	—
↓	AS-B	115.66.46.204	1.81	↓	AS-B	115.66.46.204	1.75	↑	AS-B	115.66.46.206	27.79
↓	AS-B	115.73.130.248	3.44	↓	AS-B	115.73.130.248	1.83	↑	AS-B	115.73.130.250	71.78
↓	**AS-A**	**35.12.227.158**	**2.15**	↓				↑			
↓	—	—	—	↓	—	—	—	↑	—	—	—
↓	**AS-A**	**35.12.227.158**	**34.49**	↓				↑			
↓	CDN	146.98.248.120	37.3	↓	CDN	146.98.249.115	28.7	↑	CDN	146.98.249.114	0.71
anycast	CDN	101.208.74.51	32.91	unicast	CDN	146.98.187.229	27.32	unicast	CDN	146.98.251.152	1.13
CDN											

TABLE II: An intra-AS slow hop from a VP to PoP FRA (discussion: §V-A1). Each grouped column represents a vertical traceroute record, horizontally matched with same-AS hops if any. Blanks mean no hops match with same ASes, showing routing difference.

Vantage Points											
src	**AS**	**IP**	**RTT**	**src**	**AS**	**IP**	**RTT**	**dst**	**AS**	**IP**	**RTT**
↓	—	207.213.136.212	1.38	↓	—	207.213.136.212	1.23	↑			
↓	AS-D	127.129.232.23	5.9	↓	AS-D	127.129.232.23	5.87	↑	AS-D	115.227.215.27	10.17
↓	**AS-D**	**127.129.233.106**	**5.89**	↓	AS-D	127.129.233.106	5.76	↑	AS-D	127.129.233.107	6.26
↓	**AS-E**	**35.130.33.74**	**27.68**	↓				↑			
↓	AS-E	35.130.248.142	28.2	↓				↑			
↓	AS-E	35.130.89.0	28.34	↓				↑			
↓	AS-E	211.205.94.59	33.13	↓				↑			
↓				↓	AS-F	126.82.128.149	13.39	↑	AS-F	126.82.128.201	6.28
↓	CDN	146.98.249.115	30.21	↓	CDN	146.98.249.115	12.76	↑	CDN	146.98.249.114	1.44
anycast	CDN	101.208.74.51	20.33	unicast	CDN	146.98.251.67	11.38	unicast	CDN	146.98.251.152	0.74
CDN											

TABLE III: An inter-AS slow hop from a VP to PoP FRA (discussion: §V-A2).

Vantage Points											
src	**AS**	**IP**	**RTT**	**src**	**AS**	**IP**	**RTT**	**dst**	**AS**	**IP**	**RTT**
↓	AS-K	206.13.250.110	0.43	↓	AS-K	206.13.250.110	0.39	↑	AS-K	206.13.250.111	1.99
↓				↓	AS-K	107.219.4.113	0.51	↑	—	—	—
↓	AS-L	72.58.248.181	0.78	↓				↑			
↓	—	207.122.137.141	0.5	↓				↑			
↓	—	—	—	↓				↑			
↓	**AS-H**	**7.213.227.109**	**27.1**	↓				↑			
↓				↓	—	112.154.229.152	2.64	↑	—	112.154.229.69	1.77
↓	CDN	146.98.108.78	27.12	↓	CDN	146.98.108.78	3.28	↑	CDN	146.98.108.79	3.64
anycast	CDN	101.208.74.51	25.86	unicast	CDN	146.98.110.124	2.2	unicast	CDN	146.98.110.208	0.7
CDN											

TABLE IV: A near-CDN slow hop from a VP to PoP VIE (discussion: §V-A3).

routing can be improved. Nearly all of these cases are due to slow hops (§IV-C).

For cases where we find U-probing is faster in Table VI, we are able to locate the slow hops for most (130 out of 133) of them. We enumerate all the ASes the appear as slow hops (one AS for an inter-AS slow hop, and two for intra-AS). We find 77 ASes appear in slow hops. Three of these appearing very frequently, each affecting about 20 VPs while others affecting one or two VPs. We focus remediation efforts on these three ASes, since changes there will improve service for many users. We consider each of these three cases next.

A Single Inbound Provider In the first case, an inter-AS slow-hop happens inside a large regional provider, labled AS-H . RTT increases by about 20 ms affecting about 19 VPs over time. For each of the 19 VPs, we see the increase happens at one of the three route interfaces, suggesting congestion or other challenges at three places. U-probing suggests a better route available through an alternative provider, since the traceroute from the CDN to the VP avoids the delay.

We considered several remediation options, but choices are somewhat limited because the slow hop is on the inbound path and the CDN must convince its clients to take a new path to the CDN. In the case that the slow hop occurs on a hop adjacent to the CDN, two primary routing options are available to the CDN operator. First, it could withdraw announcements to AS-H completely, but that risks leaving some clients of that AS with poor connectivity to the CDN. Alternatively, the CDN can use a community string to request its peer refrain from propagating the CDN's anycast router to AS-H 's peers. Such a change may then encourage distant clients to consider

Vantage Points											
src	**AS**	**IP**	**RTT**	**src**	**AS**	**IP**	**RTT**	**dst**	**AS**	**IP**	**RTT**
↓	AS-M	119.204.17.68	0.46	↓	AS-M	119.204.17.68	0.45	↑	AS-M	119.204.17.70	9.35
↓				↓				↑	AS-M	119.204.23.5	9.0
↓				↓				↑	AS-G	205.125.195.218	11.56
↓				↓				↑	AS-G	35.82.197.190	10.0
↓				↓				↑	AS-G	35.82.246.105	11.47
↓	AS-N	62.245.207.153	7.51	↓	AS-N	62.245.207.153	7.48	↑	AS-G	62.214.105.74	9.71
↓	AS-N	82.135.16.136	7.47	↓	AS-N	82.135.16.136	7.72	↑	AS-G	62.214.38.165	11.59
↓	~~AS-N~~	~~210.69.126.74~~	~~16.54~~	↓	~~AS-N~~	~~210.69.126.74~~	~~17.0~~	↑	AS-G	62.214.37.137	0.66
↓				↓				↑	AS-F	126.82.128.45	0.66
↓	CDN	146.98.249.115	10.23	↓	CDN	146.98.249.115	13.18	↑	CDN	146.98.249.114	1.01
anycast	CDN	101.208.74.51	22.47	unicast	CDN	146.98.251.67	9.44	unicast	CDN	146.98.251.152	0.81
CDN											

TABLE V: A circuitous path from a VP to PoP FRA (discussion: §V-A4). (The striking-out means RTT of this hop is a false slow hop (see §IV-C))

	2019-07-29	
	VPs	**Percnt.**
VPs in use	8350	100.00%
equal RTTs	7967	95.37%
consistent unequal RTTs	383	4.59%
A-probing faster	250	2.99%
U-probing faster	**133**	**1.59%**
slow hops	130	1.56%
inter-AS	51	0.63%
intra-AS	16	0.19%
either intra- or inter-AS	63	0.77%
circuitous path	3	0.04%

TABLE VI: BAUP results

scenario	**VP count**
Seen via AS-H	171
inactive after fix	33
active after fix	138
improved	130
still via AS-H	1
not via AS-H	129
got worse	8
still via AS-H	7
not via AS-H	1

TABLE VII: VPs affected by AS-H before and after fixing

alternative inbound paths. Prompted by our work, the CDN made changes to address this problem, as we describe in §VI.

Internal Routing Policy Our second case results from internal policies at the CDN which cause clients to use an indirect route over a direct peering link. §V-A1 and Table II show traceroutes for this case. Here, we observe an intra-AS slow hop happening within AS-A , a large Transit provider. The slow hop has inflation of about 20-30 ms, affecting 15 VPs over time. Each of the 15 VPs, sees the increase happening at one of two router interfaces, suggesting problems in two places. Unfortunately, existing CDN policies withholds anycast announcements from these peers, causing them to use a transit route. Addressing this case requires changes to the CDN routing policy, a topic currently under evaluation.

External Routing Policy Our final example is the result of policy determined by an external network. Here, the slow hop is within AS-J , a large regional ISP. In this case, we see a hop-to-hop RTT increases between 15 ms and 25 ms, at different router interfaces, to different 17 VPs. U-probing suggestions also vary, with lower-RTT return paths passing through a handful of other networks.

This case appears to be the result of external peering policy, outside the control of the CDN operator, which prefers certain inbound routes. Changing these policies requires inter-operator negotiation, so while BAUP cannot suggest CDN-only mitigations, it does help identify the problem. Identification helps operators detect and quantify the impacts of policies, helping prioritize potential resolution.

Circuitous Paths Circuitous paths can have solutions similar to slow hops. With only three cases (Table VI), we have not yet examined specific mitigations.

VI. Improving Performance with BAUP

Following the inbound provider example in §V-C and directly motivated by the result of BAUP, we worked with the CDN operators to adjust routing to AS-H . Changes to routing must be done carefully, because even though we expect it to improve performance for the 19 VPs found in BAUP, we must be careful it does not degrade performance to other CDN users. The change made by the CDN operators was to add a community string requesting that AS-H refrain from announcing the CDN's anycast route to its peers.

Although our goal was to improve the 19 VPs found by AU probing, in fact we found *171 VPs that pass through AS-H improve* (from traceroutes to the CDN as of 2019-07-29). BAUP finds only 19 of them because of its strict requirement for consistent, unequal RTTs, but examination shows that AS-H contains a slow hop for all 171 VPs.

After the CDN made the routing changes, we re-examined the 138 VPs (of the 171 behind AS-H) that were still active. We found *latency significantly decreased for nearly all VPs in this group*, in some cases reducing in half. Figure 2 shows latency before and after the change, with most VPs reducing latency by about 20 ms, with the median latency falling from 39.77 ms to 16.27 ms. Figure 3 shows the change in performance for each VP: we see that a few (8 of 138) show slightly larger latency, but 80% show their RTT drop by

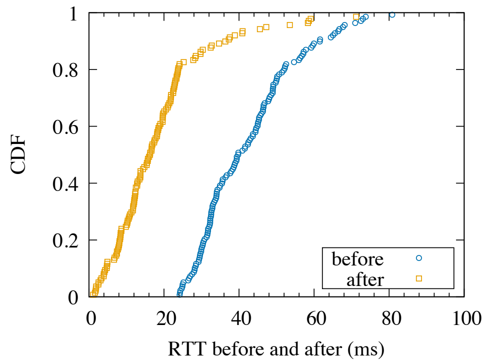

Fig. 2: CDF of RTT before and after applying fix

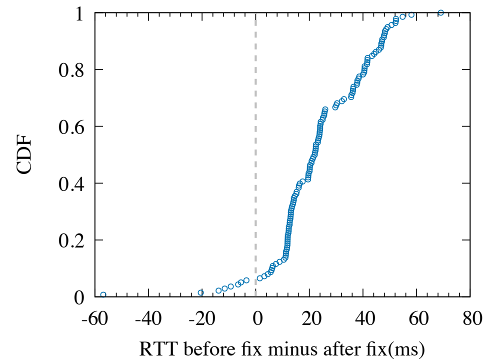

Fig. 3: CDF of RTT before minus RTT after applying fix

10 ms or more, and 55% by 20 ms or more. When we examine the 8 VPs that show higher latency, 7 of them still reach CDN via AS-H and so were not actually affected by our routing changes; we believe the last is measurement noise.

VII. Related Work

We build upon three categories of prior studies of anycast and CDN performance: first type as evaluation of overall anycast performance, second type as optimizing anycast latency, and third type as predicting end-to-end latency. Prior evaluations of anycast motivate our work by suggesting potential routing inefficiency and possibilities to improve latency, and studies of latency motivate our study of hop-by-hop paths.

Prior studies evaluated several production CDNs, each with different architectures. Google's WhyHigh found most users are served by a geographicly near node, but regional clients can widely different latencies even when served by a same CDN node [22]. Microsoft found roughly 20% of clients to a suboptimal front-end in their CDN [9]. Other work has studied the latency and geographic anycast catchment based on the root DNS infrastructure [19], [24], [31], [23] . Fontugne et al. detect network disruptions and report them in near-real-time with traceroute [20]. Our work extends theirs by using information from *both* directions and for both anycast and unicast paths, allowing us to not only find network disruptions, but also routing detours without disruptions. We also share WhyHigh's motivation to find slow hops from congestion and circuitous routing. While Li et al. found routing was often inefficient [23], we found that latency problems were relatively rare—perhaps because we focus on available network paths while they consider geographic distance and because they examine denser anycast networks, which have more opportunities for suboptimal routing. In addition, the CDN we studied employs *regional* announcements, were anycast announcements are restricted to a single continent, limiting how far latency can be off. Schmidt et al. [31] showed that additional anycast instances show diminishing returns in reducing latency, and suggest per-continent deployments (as seen in the CDN we study). Bian et al. [6] showed that 19.2% of global anycast prefixes have been potentially impacted by remote peering.

Our work emphasizes a lightweight evaluation method. WhyHigh diagnoses problem by clustering nearby clients together, and picks the shortest latency to compare, which requires the location data of clients to be precise [22] . FastRoute optimized anycast usage, using multiple addresses for different areas using a hybrid anycast with DNS-based selection [18]. Like WhyHigh, FastRoute also diagnoses latency problems based on user locations. Our work focuses on diagnosis, rather than prediction like iPlane [25], [26]. Our methodology uses the difference between routing segments in A and U Probing by a simple RTT-inequality indicator. Moreover, we don't require the precise location of each client (vantage point), or each router on path. Every time we detect, we compare from the same VP and same destination, so there is no risk of error due to an incorrect IP-geolocation mapping.

VIII. Conclusions

BAUP allows anycast CDNs to detect opportunities to improve latency to their clients due to congested links and routing detours. By comparing the route taken towards a CDN's anycast and unicast addresses, BAUP detects opportunities to improve latency, and with bidirectional traceroutes we observe slow hops and circuitous paths. We show that these observations allow BAUP to identify opportunities for improvement. Working with a CDN operator, we show that changes identified by BAUP halved latency for some VPs, affecting 91 ASes in 19 countries with more than 100k users. Since Internet routing is always changing, we suggest that BAUP should be used to test anycast deployments regularly. It can help debug performance problems and detect regressions.

Acknowledgments

The work of John Heidemann and Lan Wei in this paper is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-18-2-0280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] Akamai-Content Distribution Network: https://www.akamai.com/us/en/resources/content-distribution-network.jsp (Jan 2020)
[2] ANT/ISI: https://ant.isi.edu/software/cryptopANT/index.html (Jul 2018)
[3] APNIC: Bgp-stats routing table report—japan view. https://mailman.apnic.net/mailing-lists/bgp-stats/archive/2020/02/msg00085.html (Feb 13 2020), https://mailman.apnic.net/mailing-lists/bgp-stats/archive/2020/02/msg00085.html
[4] Austein, R.: DNS name server identifier (NSID) option. RFC 5001, Internet Request For Comments (Aug 2007)
[5] Bajpai, V., Eravuchira, S.J., Schönwälder, J.: Lessons learned from using the ripe atlas platform for measurement research. ACM SIGCOMM Computer Communication Review **45**(3), 35–42 (2015)
[6] Bian, R., Hao, S., Wang, H., Dhamdhere, A., Dainotti, A., Cotton, C.: Towards Passive Analysis of Anycast in Global Routing: Unintended Impact of Remote Peering. ACM SIGCOMM Computer Communication Review (CCR) **49**(3) (Jul 2019)
[7] Caesar, M., Rexford, J.: Bgp routing policies in isp networks. IEEE network **19**(6), 5–11 (2005)
[8] Calder, M., Fan, X., Hu, Z., Katz-Bassett, E., Heidemann, J., Govindan, R.: Mapping the expansion of Google's serving infrastructure. In: Proceedings of the ACM Internet Measurement Conference. pp. 313–326. ACM, Barcelona, Spain (Oct 2013). https://doi.org/http://dx.doi.org/10.1145/2504730.2504754
[9] Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., Padhye, J.: Analyzing the performance of an anycast cdn. In: Proceedings of the 2015 ACM Conference on Internet Measurement Conference. pp. 531–537. ACM (2015)
[10] Chiu, Y.C., Schlinker, B., Radhakrishnan, A.B., Katz-Bassett, E., Govindan, R.: Are we one hop away from a better internet? In: Proceedings of the 2015 Internet Measurement Conference. pp. 523–529 (2015)
[11] CloudFare: https://www.cloudflare.com (Oct 2019)
[12] de Vries, W.B., de O. Schmidt, R., Hardaker, W., Heidemann, J., de Boer, P.T., Pras, A.: Verfploeter: Broad and load-aware anycast mapping. In: Proceedings of the ACM Internet Measurement Conference. London, UK (2017)
[13] Dean, J., Barroso, L.A.: The tail at scale. Communications of the ACM **56**(2), 74–80 (Feb 2013). https://doi.org/10.1145/2408776.2408794
[14] Dhamdhere, A., Clark, D.D., Gamero-Garrido, A., Luckie, M., Mok, R.K., Akiwate, G., Gogia, K., Bajpai, V., Snoeren, A.C., Claffy, K.: Inferring persistent interdomain congestion. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. pp. 1–15. ACM (2018)
[15] Edgecast: https://www.verizondigitalmedia.com (Oct 2019)
[16] Fan, J., Xu, J., Ammar, M.H., Moon, S.B.: Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme. Computer Networks **46**(2), 253 – 272 (2004)
[17] Fan, X., Heidemann, J., Govindan, R.: Evaluating anycast in the Domain Name System. In: Proceedings of the. pp. 1681–1689. IEEE, Turin, Italy (Apr 2013)
[18] Flavel, A., Mani, P., Maltz, D., Holt, N., Liu, J., Chen, Y., Surmachev, O.: Fastroute: A scalable load-aware anycast routing architecture for modern cdns. In: 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15). pp. 381–394. USENIX Association, Oakland, CA (May 2015)
[19] Fomenkov, M., Claffy, K.C., Huffaker, B., Moore, D.: Macroscopic internet topology and performance measurements from the dns root name servers. In: LISA. pp. 231–240 (2001)
[20] Fontugne, R., Pelsser, C., Aben, E., Bush, R.: Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. In: Proceedings of the 2017 Internet Measurement Conference. p. 1528. IMC 17, Association for Computing Machinery, New York, NY, USA (2017)
[21] Govindan, R., Tangmunarunkit, H.: Heuristics for Internet map discovery. In: Proceedings of the. pp. 1371–1380. IEEE, Tel Aviv, Israel (Mar 2000)
[22] Krishnan, R., Madhyastha, H.V., Jain, S., Srinivasan, S., Krishnamurthy, A., Anderson, T., Gao, J.: Moving beyond end-to-end path information to optimize cdn performance. In: Internet Measurement Conference (IMC). pp. 190–201. Chicago, IL (2009)
[23] Li, Z., Levin, D., Spring, N., Bhattacharjee, B.: Internet anycast: performance, problems, and potential. In: SIGCOMM. pp. 59–73 (2018)
[24] Liu, Z., Huffaker, B., Fomenkov, M., Brownlee, N., et al.: Two days in the life of the dns anycast root servers. In: International Conference on Passive and Active Network Measurement. pp. 125–134. Springer (2007)
[25] Madhyastha, H.V., Anderson, T., Krishnamurthy, A., Spring, N., Venkataramani, A.: A structural approach to latency prediction. In: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. pp. 99–104. ACM (2006)
[26] Madhyastha, H.V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., Venkataramani, A.: iplane: An information plane for distributed services. In: Proceedings of the 7th symposium on Operating systems design and implementation. pp. 367–380. USENIX Association (2006)
[27] Marder, A., Luckie, M., Dhamdhere, A., Huffaker, B., kc claffy, Smith, J.M.: Pushing the boundaries with bdrmapIT: Mapping router ownership at Internet scale. In: Proceedings of the ACM Internet Measurement Conference. ACM, Boston, Massachusetts, USA (Oct 2018)
[28] RIPE NCC: RIPE Atlas measurement about vdms. https://atlas.ripe.net/measurements/ID (2019), https://atlas.ripe.net/measurements/, ID: towards anycast: 22385847, towards unicast: contact the paper authors for RIPE experiment IDs.
[29] Root Operators: http://www.root-servers.org (Feb 2020)
[30] Route Views: http://archive.routeviews.org/ (2019)
[31] Schmidt, R.d.O., Heidemann, J., Kuipers, J.H.: Anycast latency: How many sites are enough? In: Proceedings of the Passive and Active Measurement Workshop. pp. 188–200. Springer, Sydney, Australia (Mar 2017)
[32] de Vries, W., Santanna, J.J., Sperotto, A., Pras, A.: How asymmetric is the internet? In: IFIP International Conference on Autonomous Infrastructure, Management and Security. pp. 113–125. Springer (2015)
[33] Wei, L., Heidemann, J.: Does anycast hang up on you (udp and tcp)? IEEE Transactions on Network and Service Management **15**(2), 707–717 (Feb 2018)