Tracing Anycast: a Performance Assessment

Paper #A, B pages body, C pages total

ABSTRACT

Anycast allows for providing services from multiple, geographically distant, Points of Presence (PoPs) using a single IP address to *e.g.*, provide increased resilience. However, due to its opaqueness it is often unknown which addresses are provided using anycast and, if so, where its PoPs are located. As anycast is widely used for *e.g.*, critical Internet infrastructure like the DNS, there have been efforts in mapping anycast deployments. The current state-of-the-art, a technique called iGreedy, relies on latency measurements. However, such measurements are noisy as they suffer from *e.g.*, network propagation delays.

Previous work has shown that traceroute, a technique that reveals the individual hops a packet traverses, can be used to detect anycast. Furthermore, it can locate anycast using available geolocation data of hops near anycast PoPs.

This work is the first to assess the performance of the traceroute technique at scale by performing it towards O(4) anycast prefixes. Our results show ... We provide an extensive validation of the traceroute technique, publicly release our code, and release an extensive geolocation dataset of anycast networks.

1 INTRODUCTION

Anycast is the practice of making an Internet address available in multiple discrete locations [4]. This allows operators to offer services closer to clients and increase resilience through redundancy. Examples include DNS resolvers and nameservers, and CDNs providing web caching. Whilst the number of anycast addresses on the Internet are small (<1%) it has a large share in Internet traffic [1].

For this reason, efforts have been made to map anycast deployments on the Internet to *e.g.*, measure Internet resilience. The most notable are MAnycast[2] [7] and iGreedy [2] that can differentiate between unicast and anycast addresses , the latter of which can also enumerate and geolocate the Points of Presence (PoPs) behind anycast addresses. iGreedy geolocates using latency measurements by probing an anycast address from multiple geographically distributed Vantage Points (VPs), similar to conventional IP unicast geolocation. However, such latency measurements are known to be noisy due to *e.g.*, network propagation delays.

Unicast IP geolocation often uses tools like traceroute to improve precision. Traceroute allows for measuring the path that an Internet packet takes to reach its target. When determining the location of a target, traceroute can be used to find nearby hops that have available location information. It has also been used to geolocate anycast using the location of hops near PoPs, to infer the location of the PoPs themselves. One example is verification of GDPR compliance [6]. However, traceroute has limitations like tunneling that keep hops hidden and may result in indeterminate geolocation results.

This work puts the traceroute technique to the test by performing it at scale towards 13k anycast prefixes. Our contributions are i) an extensive validation of the technique, ii) release of the measurement and analysis code, iii) public release of the dataset revealing up to N anycast PoPs for Y prefixes.

The paper is structured as followed. First, in §2 we briefly discuss background on anycast detection and traceroute and provide related work on using traceroute to detect anycast. Then, we detail the methodology used in §3 followed with an analysis of results in §results. Finally, we discuss the performance of traceroute to detect anycast and list further use cases in §5.

2 BACKGROUND AND RELATED WORK

In this section we first provide a background on anycast detection and traceroute. Next, we discuss related work on measuring anycast with traceroute.

2.1 Anycast detection

The current state-of-the-art in detecting anycast is iGreedy. This approach uses the speed of packets in fibre optic cables (roughly two thirds the speed of light), combined with measured ping latencies, to determine the maximum distance travelled from multiple VPs in distinct locations. In the case of a unicast target, all circles will overlap providing a single solution within the intersection of all areas (*i.e.*, trilateration).

However, in the case of an anycast target you will observe non-intersecting sets of circles where each set of circles reaches a different anycast PoP. In this case, the iGreedy algorithm finds the minimum set of independent overlapping areas in which anycast PoPs must be located for there to be no speed-of-light violation. Next, it tags the location as the largest metropolitan area within this area, as anycast operators often deploy their PoPs in such areas to maximize utility. Its output is the airport nearest to that metropolitan area.

The accuracy of this method is dependent on the number and the geographical diversity of VPs used. It has shown to be quite accurate achieving a recall N% of with an average error of N kilometers. However, it suffers from a large probing cost.

MAnycast^2 is a lightweight detection technique for anycast, requiring only a few probes to determine whether an address is anycast. This is achieved by measuring anycast using anycast, where ping packets are sent to a target from all PoPs of a measurement anycast infrastructure using an anycast source address. If the target is unicast, this unicast target receives the same probe from each anycast PoP and sends back replies for each received probe. These replies target the anycast address that was the source of the original probe, and will therefore route to the nearest PoP of the measurement infrastructure. But, if the target is anycast, the probe replies will reach different target anycast PoPs where each will send back its probe replies to different measurement infrastructure PoPs. Therefore, if a single measurement PoP receives probe replies it is inferred to be unicast and if multiple measurement PoP receive probe replies it is inferred to be anycast.f

Unfortunately, this approach will in rare cases falsely classify a unicast target as anycast when its replies reach multiple PoPs, which happens due to *e.g.*, route flaps. Therefore, the output addresses of MAnycast^2 (where more than one PoP received replies) are considered *candidate anycast* as it overestimates.

In this work, we use MAnycast^2 to generate a set of candidate Anycast Targets (ATs). These ATs are measured using the traceroute technique. Additionally, they are measured with iGreedy to provide a comparison of both anycast geolocation methodologies.

2.2 Traceroute

Traceroute traces Internet paths by triggering routers on-path towards a target to send back ICMP `TTL-exceeded` replies. Such routers are often configured with PTR records that operators use for debugging purposes, and contain information like the ASN, network type, country and city where the router is located. Thereby, it is possible to infer hints on the geographical path a packet took towards a target by looking at PTR records or geolocation database entries for such router addresses. Furthermore, it is possible to infer locations of Internet addresses using traceroute as there may be geolocation information available for nearby hops (where nearby is determined with RTT). In particular, the pen-ultimate hop (*p-hop*), *i.e.*, the hop before the destination, is often used as it is closest to the destination.

Several works use traceroute to perform unicast geolocation in this work we utilize it to geolocate anycast PoPs.

2.3 Measuring anycast with traceroute

First, in 2013 Xun et al. analysed the usage of anycast in the DNS [3]. They used CHAOS records, a record set by operators to identify the specific nameserver reached, to enumerate DNS anycast deployments. However, as unique CHAOS records were used for multiple load-balanced nameservers at a single anycast PoP they augmented it with traceroute to resolve ambiguities.

That same year, RIPE NCC released *ipmap*, a project to geolocate addresses using historical traceroute data from RIPE Atlas VPs that returns possible locations of an IP address [5]. Later, they added an anycast classification where multiple available locations would be returned.

Next, Wei et al. in 2017 measured the occurrence of anycast flipping (*i.e.*, a single client flipping between multiple anycast PoPs) where they used the pen-ultimate hops of traceroutes to detect a client reaching multiple distinct PoPs [8].

Recently, in 2023 traceroute was used to geolocate the PoPs of regional anycast deployments by Zhou et al. [9] By geolocating the p-hop as observed from RIPE Atlas VPs, they infer the location of the PoP reached by mapping it to the closest PoP location from available ground truth. They performed their methodology towards regional anycast prefixes from two CDNs and showed they were able to infer the reached site in the majority of cases.

Finally, Pascual et al. introduced a tool to trace anycast communications (*Hunter*), to verify compliance to data protection regulations for personal data transfers, in 2024 [6]. Their method geolocates the p-hop visible in a traceroute, then performs geolocation using latency measurements from nearby RIPE Atlas VPs. Their method showed high accuracy and precision in geolocating the PoP reached when validating using two anycast addresses from Cloudflare.

Unlike previous work, that used traceroute towards a select few anycast operators, we explore the effectiveness of using traceroute to geolocate anycast PoPs towards 13k anycast prefixes found using MAnycast^2 and iGreedy. By doing so, we aim to provide an extensive dataset containing the geographical distribution of these anycast networks.

3 METHODOLOGY

This work builds on the methodology developed by Zhou et al. [9].

3.1 Previous method

This would basically describe that we build on the methodology from the regional anycast paper. They use the location of hops nearby the anycast site reached that is determined with PTR records, RTT differences between hops, and ipinfo for nearby hops as a last resort.

3.2 Automating PTR record translation

Since we have to perform this as scale, and can't manually figure out the PTR record location, we intend on adding hoiho (Matthew's PTR to location script) and the new LLM variant of hoiho that Raffaele shared.

3.3 Measuring distance from traceroute hops

we also would investigate using traceroute hops as additional VPs for the iGreedy algorithm, if they show stable RTT values and have 'known' locations.

3.4 Measurement set-up

We use Ark with this many VPs that support traceroute. We should also consider using RIPE Atlas in addition. We target this candidate set created by MAnycast2. We compare with iGreedy results ran on the same set, with the same VPs. We describe ground-truth sources for validation.

4 RESULTS

4.1 Detection

How does the technique perform in detection? Does it have FPs? Does it have FNs? How does it compare to iGreedy and MAnycast2.

4.2 Enumeration

How many sites can be discovered with the traceroute technique? What does the distribution look like? How does this compare to iGreedy results? What about ground truth (using e.g., CHAOS records, public info, operator GT).

4.3 Geolocation

Similar to above.

How close does traceroute get to the real location? Does it get closer than iGreedy?

4.4 Locations

We could write about the locations found. For example, where are anycast operators often located? Where are anycast operators often not located? Do we see different deployment strategies by operators? How many sites do we find on average?

Traceroute results may also reveal where the PoPs are hosted. E.g., are they hosted by Vultr? Or Equinix? Mapping AS diversity of anycast prefixes is an interesting analysis. Also gives data on upstreams of anycast prefixes.

4.5 MAnycast2 FPs

I suspect this will shed light on the MAnycast2 FPs. These are mostly Microsoft prefixes that are announced globally, but route to unicast servers. Traceroute will reveal this, and we can further investigate or write it as future work to map this TE practice by global ASes.

5 DISCUSSION

5.1 Future work

This would discuss future work. In particular, using traceroute to look at anti-DDoS announcements, global BGP (e.g., enumerating ingress PoPs for large global ASes), running this using historical traceroute data, ...

5.2 How does traceroute perform?

Then we would discuss the results. I expect it to outperform iGreedy for some prefixes but be 'useless' for others.

We would also discuss the large number of probes needed to do traceroute and if the additional probes are worth the increased coverage.

6 ACKNOWLEDGEMENTS

kc would like us to acknowledge the GMI grant.

REFERENCES

[1] Danilo Cicalese, Danilo Giordano, Alessandro Finamore, Marco Mellia, Maurizio Munafò, Dario Rossi, and Diana Joumblatt. 2021. A First Look at Anycast CDN Traffic. arXiv:1505.00946 [cs.NI] https://arxiv.org/abs/1505.00946

[2] Danilo Cicalese, Diana Joumblatt, Dario Rossi, Marc-Olivier Buob, Jordan Augé, and Timur Friedman. 2015. A fistful of pings: Accurate and lightweight anycast enumeration and geolocation. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2776–2784. https://doi.org/10.1109/INFOCOM.2015.7218670

[3] Xun Fan, John Heidemann, and Ramesh Govindan. 2013. Evaluating anycast in the domain name system. In *2013 Proceedings IEEE INFOCOM*. 1681–1689. https://doi.org/10.1109/INFCOM.2013.6566965

[4] Kurt Erik Lindqvist and Joe Abley. 2006. Operation of Anycast Services. RFC 4786. https://doi.org/10.17487/RFC4786

[5] RIPE NCC. 2025. IPmap. https://ipmap.ripe.net/. [Accessed 31-03-2025].

[6] Hugo Pascual, Jose M. del Alamo, David Rodriguez, and Juan C. Dueñas. 2024. Hunter: Tracing anycast communications to uncover cross-border personal data transfers. *Computers & Security* 141 (2024), 103823. https://doi.org/10.1016/j.cose.2024.103823

[7] Raffaele Sommese, Leandro Bertholdo, Gautam Akiwate, Mattijs Jonker, Roland van Rijswijk-Deij, Alberto Dainotti, KC Claffy, and Anna Sperotto. 2020. MAnycast2: Using Anycast to Measure Anycast. In *Proceedings of the ACM Internet Measurement Conference* (Virtual Event, USA) *(IMC '20)*. Association for Computing Machinery, New York, NY, USA, 456–463. https://doi.org/10.1145/3419394.3423646

[8] Lan Wei and John Heidemann. 2018. Does Anycast Hang Up on You (UDP and TCP)? *IEEE Transactions on Network and Service Management* 15, 2 (2018), 707–717. https://doi.org/10.1109/TNSM.2018.2804884

[9] Minyuan Zhou, Xiao Zhang, Shuai Hao, Xiaowei Yang, Jiaqi Zheng, Guihai Chen, and Wanchun Dou. 2023. Regional IP Anycast: Deployments, Performance, and Potentials. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 917–931. https://doi.org/10.1145/3603269.3604846