Geolocating anycast IP addresses using traceroute

Chris Josten
University of Twente
Enschede, Overijssel, NL

ABSTRACT

Anycasting—mapping multiple hosts to a single IP address—has been in use for decades. It used by DNS and Content Delivery Networks for coarse load balancing, reduced latency and resilience against Denial of Service attacks. However, its deployment is hard to measure due the nature of its implementation being opaque to IP hosts on the network. In the last 15 years, methods for detecting anycast sites have started to appear. The current state of the art, iGreedy, is able to geolocate anycast addresses, but it is not able to detect all locations. This research proposes an alternate method based on the geolocation of penultimate hops in traceroutes to anycast addresses, to attempt to improve the number of locations and the accuracy. It is based on the hypothesis that for network traffic to reach an anycast site, it will need to pass through a router nearby with an unicast IP address. Since there already exist IP to geolocation databases for unicast addresses, this should give an approximate geolocation for an anycast site. The results of this method are then compared to the result of iGreedy. It is found that traceroute is significantly less accurate than iGreedy at geolocating, but it is able to detect more locations.

CCS CONCEPTS

• **Networks** → **Naming and addressing**.

KEYWORDS

Delay Measurement, Anycast, IP geolocation, Traceroute

ACM Reference Format:
Chris Josten. 2024. Geolocating anycast IP addresses using traceroute. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (41TScIT).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the context of network routing, unicast is when a single source communicates with a single destination. Another method is anycast, which allows multiple hosts providing the same network services to share the same IP address, where it does not matter to clients which hosts they access. The motivation for its development in 1993 was for simplifying service discovery [12], but its use has been expanded to provide coarse load balancing, reduced latency and resilience against Denial of Service (DoS) attacks [10]. Starting with DNS root servers, more and more services are anycasted to benefit from these perks, such as DNS servers [4] and Content Delivery Networks (CDNs).

Since anycast is beginning to be a more common occurrence, there the desire to get more insight in detecting anycast traffic there is and where the traffic is headed is increasing. Networkers are interested in understanding how traffic moves through their network because it is a big part of their business, business users of anycast services may be interested in solving issues with anycast traffic. and the information of where the hosts of anycast are located could be interested to other researchers in other areas. [5]

However, anycast is not trivial to measure. While IPv6 dedicates a subnet to anycast IP addresses, many IPv6 anycasted addresses are still using a regular IPv6 address. IPv4 does not have a dedicated subnet for anycast IP addresses. To be able to observe which addresses are anycast, one must either get hold on BGP routing data or observe traffic from multiple sources themselves.

Following the increased wishes for more insight, this has lead to research in detecting anycast. Many share a similar setup: multiple servers are used spread around the world as so-called *Vantage Points* (VP)s. From these VPs, the targeted, potential anycast IP address is probed. The first methods were mainly targeted at DNS servers, since those were early adopters of anycast. For DNS servers there is an informal convention of the CHAOS records, which can identify the server. If those identify differently when probed from other vantage points, those are most likely different servers sharing an anycasted IP address.[7].

Before getting into geolocating anycast addresses, take a look at ways of geolocating unicast addresses. Geolocating unicast IP addresses has been studied for some while now. Companies have a financial incentive to geolocate users, because they may be able to serve their users better or show targeted advertisements using their location.

One of the earliest methods available for geolocating unicast addresses, *GeoPing*, works by measuring the time it takes to for ping packets sent from multiple geographically spread out VPs with known geolocations to arrive at an IP address with unknown geolocation and considering the VP with the lowest ping the nearest location [13]. *Constraint-Based Geolocation (CBG)* improves on this putting a constraint in a form of a circle around a VP where the radius is the maximum distance the packet could have travelled assuming it travelled at the speed of light. The location can then be determined by the intersection of all circles [8]. *Topology-Based Geolocation (TBG)* adds additional constraints derived from network topology and latency data obtained from the tool ‘traceroute‘ to further narrow down the location [9].

Geolocations of IP addresses can be estimated using IP Interpolation as well: if two IP addresses have the same known geolocation and they are part of the same subnet, then a third IP address in that subnet is usually located near the first two. These techniques can all be combined to achieve a higher accurate IP to geolocation dataset

41TScIT, July 05, 2024, Enschede, Overijssel

[6]. Furthermore, there exist multiple commercial IP to geolocation databases such as MaxMind [11] and IP2Location [3], although their methodology of compiling these databases are unknown.

1.1 Problem statement

The problem this paper aims to solve is, for a given anycast IP address, to determine all of its geolocations.

When using traceroutes, it must be taken into account that not all hops are visible, meaning those do not respond with an IP address. Furthermore, some hops may respond with a non-public routable IP address, which are impossible to geolocate.

For this research, three assumptions will be made. First, whether an IP address is unicast or anycast can already be determined using other methods, to which access is provided. Secondly, unicast addresses can be geolocated. Thirdly, traceroutes from several vantage points to anycast addresses are available.

Using these assumptions, the following research questions will be answered:

(1) To which extent is it possible to use the geolocations of nearby hops of an anycast IP address as an approximate for the geolocations of anycast IP address?
   (a) How often are the nearby hops of an anycast IP address unicast addresses?
   (b) How does using the geolocations of nearby hops of an anycast IP address as an approximate compare to other, existing methods of determining the geolocations of an anycast IP address?

In this paper, first existing works will be explored, afterwards the methodology will be described, followed by the results and ending with discussion and a conclusion.

2 RELATED WORK

Some work on anycast has been done earlier in the realm of detecting anycast sites.

Fan et al. introduced a method for detecting anycast usage exclusively on DNS servers, by querying CHAOS records from *Vantage Points* (VPs). However, a limitation is that CHAOS records are not always present. To overcome this limitation, traceroute is used to identify the routers at the penultimate hop. [7]

To generalise the method to any anycast site, Cicalese et al. introduced a method based on latency for detecting anycast instances named iGreedy. If an anycast IP is detected, it will attempt to enumerate all anycast sites behind the address and in an iterative way determine the locations of all sites. However, it often underestimates the amount anycast instances and since it is delay-based, small fluctuations in lantency can be amplified in the geolocating process.[5]. It is the only method that can geolocate any anycast IP sofar, irregardeless whether the instances run specific software such as a DNS server.

For making detecting each anycast site in the IPv4 address space feasible, Sommese et al. proposed another method named MAnycast2. It works by sending echo packets from multiple VPs with the same anycast IP address and counting how many distinct VPs received a reply. It is limited to only detecting whether an IP is anycast, it cannot geolocate anycast addresses.[14].

3 METHODOLOGY

In this section, the methodology of the research will be described. First the measurement setup and dataset used are described, next the method for answering research question 1a and finally the method for research question 1b.

Traceroutes from 32 geographically distributed vantage points to 13 DNS root servers were collected before the start of this research, making for 416 traceroutes total. A full list of vantage points is available in Appendix A. The DNS root servers were selected as target since the actual locations of these servers are published by the Root Server Technical Operators Association. These locations will be referred to as the true location. To determine the location of unicast addresses an IP to geolocation database will be used, specifically the MaxMind GeoIP database[11] will be used.

3.1 Classifying penultimate hops

First, for each traceroute and probe, the nearby hop is determined and traceroute probes that never reach the destination or have no nearby hop. The nearby hop is considered the last hop before the destination that is geo-locatable, meaning it must fulfil the following two selection criteria:

(1) The hop must be visible
(2) The IP address of the hop must not be in the IANA Special Purpose IPv4 Address Registry [1]—which would make it a bogon hop
(3) The IP address of the hop must be unicast

To answer research question 1a, all probes in each traceroute that do not fulfil criteria 1 and 2 are deleted. Then, for the remaining nearby hops, the amount that fulfils criteria 3 is calculated to answer the research question. An IP address is considered to be anycast if it is contained within the MAnycastR census[2].

3.2 Compare geolocations to other methods

To answer research question 1b on how well traceroute performs compared to other methods, the locations found by traceroute are compared to iGreedy and the true location. To do this, first the error in distance from the locations traceroutes to the true location is calculated and second the same is done for locations obtained using iGreedy.

First, for all DNS root servers, the nearby hops as described in the previous section are collected and their geolocations are looked up in the IP to geolocation database. Since up to three probes are performed per traceroute from a single vantage point to a DNS root server, the probe with the nearby hop that is located geographically the closest to the true location is then taken as the location for that traceroute. Next, the distance from the traceroute location to the true location is then calculated using the haversine formula, to take into account that the earth is a spheroid.

Secondly, for iGreedy, the MAnycastR census of anycast instances and their locations is used. Despite what the name may suggest, the geolocations are obtained via iGreedy. Here as well, the error distance from each location found in the census to the true location is calculated using the haversine formula.

Finally, the error distances of both methods are plotted in a CDF plot. The amount of distinct nearby hop addresses are counted as well, since those represent the amount of locations that are found.

Traceroute usability		Classification	
Unusable	204	Destination not reached	185
		No usable hops	19
Usable	212	Nearby hop unicast	212
		Nearby hop anycast	0
Total			416

Table 1: Usable traceroutes by selection criteria

DNS root	Actual count	Traceroute count	iGreedy count
A	59		10
B	6		7
C	12	23	7
D	209	29	18
E	328	15	28
F	345	4	24
G	6		
H	12	21	8
I	82		11
J	150	13	16
K	120	19	12
L	151	17	6
M	20	14	5

Table 2: Count of anycast instances detected

4 RESULTS

4.1 Classifying penultimate hops

The results of the selection criteria can be seen in Table 1. There are two things that stand out: firstly, about half of the traceroutes are unusable and secondly, all nearby hops are unicast.

The first observation is unfortunate, but not entirely unexpected. A large amount never reaches the destination IP address. This could be caused by a network operator detecting and blocking traffic from the traceroute tool. For traceroutes that do reach the destination, but there are no usable hops in between, there could be two cases: all hops in between did not meet the selection criteria or there were no hops between the vantage point and the destination.

The second observation is as expected: most network traffic passes through a router and does not explicitly address it, so an anycast IP address offers little if no benefit over an unicast address.

4.2 Comparison against other methods

First, the error in distance is compared, then the amount of detected locations. The distance as obtained as described in the above paragraph are shown in the following Figure 1. It should be noted

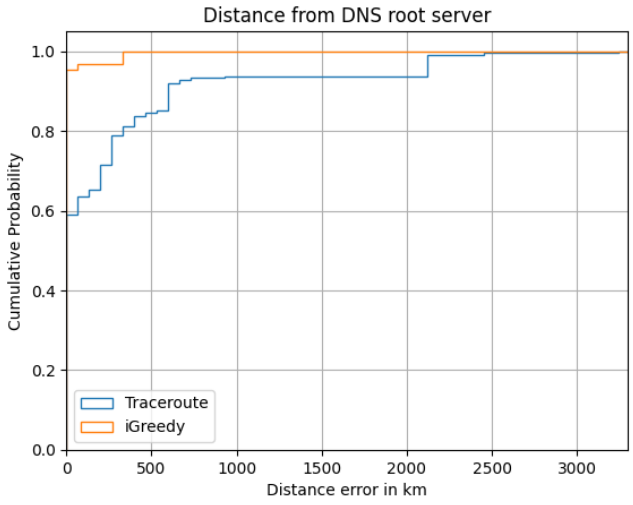

Figure 1: Comparision of error of distance between the true location and traceroute and iGreedy

that for root servers A, B, G and I there are no usable traceroutes because they did not satisfy the selection criteria, since the traceroutes never reached the root server. There was no time to figure out why it was the case for especially these servers.

In general, traceroute performs worse than iGreedy. A possible reason is that the penultimate hop is not always selected, since if it does not meet the selection criteria, the previous hop that does is selected. Some hops are located far from each other geographically but are directly connected to each other, for example the cables undersea from Europe to America. If on a traceroute from Europe to America the hops on the American side would be invisible, it could lead to a significant increase in distance to the destination.

Furthermore, the amount of locations detected can be seen in table 2. In general, traceroute is able to detect more locations than iGreedy. However, traceroute is detecting too many locations, for example for DNS root server C. This can be explained by that there are multiple routers in front of a single server, since the IP addresses of these nearby hops seem to be from the same subnet. However, it does not always detect more locations than iGreedy. This could be caused by too many results being cut by the selection criteria, for example for DNS root server F. The IP addresses of the nearby hops do not seem to have any subnet in common.

The actual count is in almost all cases larger than both the traceroute and iGreedy count. This is a result of the relatively low amount of vantage points compared to the amount of DNS root servers behind a single anycast IP.

5 DISCUSSION

Several improvements could be made to improve the accuracy of the method in this paper. One improvement that was omitted due to time constraints is to consider the latency information of traceroute of the penultimate hop and ultimate hop. If the difference in latency exceeds a certain threshold, the argument could be made that the

penultimate hop lies too far away from the ultimate hop to be used as an suitable approximate for the location of the ultimate hop.

Another accuracy improvement could be discarding traceroutes where the penultimate hop does not meet the selection criteria instead of picking the last hop that is not the destination, which does meet the selection criteria. Picking any hop that is not the penultimate hop will often result in a greater distance to the destination, at the cost of selecting even less traceroutes for inclusion.

To increase the amount of selected traceroutes, traceroutes with no hops between the vantage point and the destination could be picked. The vantage point can then act as the penultimate hop in the case the destination, in the case located in the same data centre.

Perhaps traceroute can augment iGreedy similarly traceroute is used in the CHAOS method: to disambiguate between destinations that are located close to each other, which is one of the weaknesses of iGreedy. While the distances that traceroute provides can be way of, it is able to detect more anycast instances.

6 CONCLUSION

In this paper, we took a look at geolocating anycast IP addresses using a nearby hop of the destination obtained from traceroutes. The error in distance between those locations and the true locations of the destinations of the DNS root servers were then compared to the error in distance of the locations obtained via iGreedy. The traceroute method is significantly worse than iGreedy

It can be seen that the method outlined in this paper performs worse than the already existing iGreedy method. Many traceroutes are unusable and rejected by the selection criteria. In all occasions, it performs worse than the already existing iGreedy method. It is able to detect more instances, but the accuracy of that has not been measured properly.

REFERENCES

[1] 2009. *IANA IPv4 Special-Purpose Address Registry.* Technical Report. IANA. https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml

[2] 2024. *Anycast Census.* Retrieved 2024-05-28 from https://github.com/anycast-census/anycast-census

[3] 2024. *IP Address to IP Location and Proxy Information.* Retrieved 2024-05-13 from https://www.ip2location.com/

[4] 2024. *Root Server Technical Operations Association.* Retrieved 2024-05-11 from https://root-servers.org/

[5] 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 ISSN: 0743-166X.

[6] Ovidiu Dan, Vaibhav Parikh, and Brian D. Davison. 2021. IP Geolocation Using Traceroute Location Propagation and IP Range Location Interpolation. In *Companion Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 332–338. https://doi.org/10.1145/3442442.3451888

[7] 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 ISSN: 0743-166X.

[8] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. 2004. Constraint-Based Geolocation of Internet Hosts. (2004). https://doi.org/10.1145/1028788.1028828

[9] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC '06)*. Association for Computing Machinery, New York, NY, USA, 71–84. https://doi.org/10.1145/1177080.1177090

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

[11] MaxMind. 2024. *MaxMind: Industry leading IP Geolocation and Online Fraud Prevention.* Retrieved 2024-05-13 from https://www.maxmind.com/en/home/

[12] Trevor Mendez, Walter Milliken, and Craig Partridge. 1993. *Host Anycasting Service.* Technical Report RFC 1546. https://doi.org/10.17487/RFC1546

[13] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. An investigation of geographic mapping techniques for internet hosts. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01)*. Association for Computing Machinery, New York, NY, USA, 173–185. https://doi.org/10.1145/383059.383073

[14] 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 (IMC '20)*. ACM. https://doi.org/10.1145/3419394.3423646

A VANTAGE POINT LOCATIONS

Country	City
Australia	Melbourne
	Sydney
Brazil	São Paulo
Canada	Toronto
Chile	Santiago
France	Paris
Germany	Frankfurt
India	Bangalore
	Delhi
	Mumbai
Israel	Tel Aviv-Yafo
Japan	Osaka
	Tokyo
Mexico	Mexico City
Netherlands	Amsterdam
Poland	Warsaw
Sweden	Stockholm
Singapore	Singapore
South Africa	Johannesburg
South Korea	Seoul
Spain	Madrid
United Kingdom	London
	Manchester
USA	Atlanta
	Chicago
	Dallas
	Honolulu
	Los Angeles
	Miami
	New York
	San Francisco
	Seattle

B USAGE OF AI TOOLS

No AI tools have been used in writing this report. The free and open source version of LanguageTool has been used to check for spelling and grammar mistakes, but no AI features were used.

For programming, the line completion feature of JetBrains PyCharm was used, which uses a local deep learning model.