# IP Geolocation Using Traceroute Location Propagation and IP Range Location Interpolation

Ovidiu Dan*
Microsoft Bing
Redmond, WA, USA
contact@ovidiudan.com

Vaibhav Parikh
Microsoft Bing
Redmond, WA, USA
vparikh@microsoft.com

Brian D. Davison
Lehigh University
Bethlehem, PA, USA
davison@cse.lehigh.edu

## ABSTRACT

Many online services, including search engines, content delivery networks, ad networks, and fraud detection utilize IP geolocation databases to map IP addresses to their physical locations. However, IP geolocation databases are often inaccurate. We present a novel IP geolocation technique based on combining propagating IP location information through traceroutes with IP interpolation. Using a large ground truth set, we show that physical locations of IP addresses can be propagated along traceroute paths. We also experiment with and expand upon the concept of IP range location interpolation, where we use the location of individual addresses in an IP range to assign a location to the entire range. The results show that our approach significantly outperforms commercial geolocation by up to 31 percentage points. We open source several components to aid in reproducing our results.

## CCS CONCEPTS

• **Information systems → Location based services**; • **Networks → Location based services**; • **Social and professional topics → Geographic characteristics**.

## KEYWORDS

IP geolocation, geographic targeting, geotargeting, geographic personalization, traceroute, IP range, location interpolation, IP interpolation

**ACM Reference Format:**
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 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3442442.3451888

## 1 INTRODUCTION

Online services such as search engines determine the location of users at city-level granularity by consulting IP geolocation databases, which map IP ranges to physical locations. This location information is then used for **geographic personalization**. For example, the generic query *weather* does not contain an *explicit* location. In order to serve an answer with the local forecast, the search engine needs to determine the *implicit* location of the user. Global positioning sensors can provide the precise location of the users, if they have opted-in to location sharing and if their devices contain the necessary hardware. However, in the vast majority of cases, the exact location is not available if users are using a PC without GPS hardware, or if they opt-out of location sharing. In this case, the search engine falls back to using the IP address of the device to determine a coarse location using a commercial IP geolocation database.

Commercial geolocation services such as MaxMind [27], Neustar IP Intelligence [29], and IP2Location [14] are considered state of the art, although the exact methods they use are proprietary. Recent work has questioned their accuracy [8, 32, 35]. Despite their shortcomings, IP geolocation databases are used in many other applications, including content personalization and online advertising to serve local content [11, 18], content delivery networks to direct users to the closest datacenter [12], law enforcement to fight cybercrime [36], geographic content licensing to restrict content streaming by region [25], and e-commerce to display variable pricing based on local taxes and shipping [37].

**Our work focuses on using latency differences along the traceroute path, combined with interpolation at the IP range level, to improve IP geolocation**. We base our approach on two assumptions. Our first assumption is that nodes which are close together in terms of latency on a traceroute path are also near in terms of geographic distance. Our second assumption is that addresses in a consecutive IP range are often located in the same geographic region. We evaluate to what degree these two assumptions are true using a large ground truth set of 8.9 million IP addresses, one of the largest ever reported in literature. We then combine and expand upon these two hypotheses to improve IP geolocation. More specifically, our contributions are:

(1) In our preliminary investigation we define the concept of latency neighbors. We show that there is a direct relationship between latency differences along the traceroute path and physical distance in kilometers. We propose to exploit this property to improve IP geolocation. We also investigate geographic colocation of IP range addresses. We demonstrate that when two IPs are in the same range, they are also very often located in the same geographic region. Based on this finding, we propose the concept of IP range interpolation. We find that if two or more IPs in an IP range are in the same location, then it is likely that all IPs in the range are also in that location.
(2) We propose combining the concepts of traceroute latency neighbors and IP range interpolation to improve IP geolocation. We interpolate locations from the training set to increase its coverage.

[*] Research carried out as part of doctorate studies at Lehigh University.

*WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia*

ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3451888

Then, we use latency neighbors to further propagate locations from IP ranges with known locations to IP ranges with unknown locations, using traceroute latency neighbors in aggregate. We evaluate our approach against two state of the art commercial IP geolocation databases, using a large traceroute dataset and a large ground truth set. We show that our approach significantly outperforms two commercial geolocation databases.

**Although traceroutes, IP colocation, and IP interpolation have been studied before, to our knowledge this is the first time that they have been combined together using this precise approach.** We discuss the differences to past work in more detail below. It's also important to understand that this proposal is part of a larger IP geolocation project where we combine multiple approaches [3–5]. The work presented here is only one of these proposals, and it itself is not sufficient to entirely solve the problem of creating a new IP geolocation database. However, our proposal can be used to augment and improve existing databases.

## 2 RELATED WORK

We divide relevant IP geolocation research in three broad categories, depending on the methods they use: *network delay* approaches use ping, *network topology* approaches further utilize traceroute and BGP network structure information, and *IP interpolation* approaches try to estimate the location of addresses by using information from numerically nearby IPs. Finally, we also briefly mention other geolocation approaches that use Internet data mining.

### 2.1 Network Delay

Most IP geolocation research relies on active network delay measurements (ping) to locate addresses. Early work on IP geolocation by Padmanabhan and Subramanian discusses *GeoPing* [30], which sends ICMP packets from geographically distributed landmark servers to the target IP. It then assigns the target IP the location of the closest landmark server in terms of latency.

*CBG* [9] goes further by drawing circles on the surface of the Earth around each landmark server, where the radius of each circle is given by the network delay. It then picks the center of the intersection of these circles as the likely location of the target IP. This technique is called multilateration. To estimate the conversion between network delay and physical distance, CBG builds a simple model for each probe server by measuring the delay to all the other servers. Since the locations of all probe servers is known, a *bestline* that fits below all measurements can then be determined. Later work based on CBG focuses on ways to better determine the bestline. Youn et al. [39] propose a statistical method called Statistical Geolocation, or *SG* for short, to better estimate the bestline by applying kernel density estimation to delay measurements. *Spotter* [20], a technique proposed by Laki et al., further refines the task of fitting a baseline. Instead of generating a separate bestline for each server, Spotter derives a single bestline for all of them, by combining readings from all active probes together into a common delay-distance model. Dong et al. propose *SDP*, which clusters the readings on the delay-to-distance graph by using k-means clustering. For a given target IP, it first picks the closest cluster and then it performs more extensive probes from the subset of probing nodes local to that cluster [6].

Although incremental, more recent work by by Khan et al. extends CBG by introducing a similar two-step process. In the first step they identify a coarse region using worldwide probe servers, and then in the second step they further refine the location using regional servers. Jiang et al. also propose a very similar two tiered approach that employs a radial basis function (RBF) neural network [16].

Ciavarrini et al. have recently demonstrated that network delay approaches have a best-case error of 20 kilometers and that obtaining an error below this threshold requires a number of active measurement servers so large as to be unpractical [2].

**Our approach significantly differs from traditional delay measurement geolocation approaches** that use landmark-based multilateration [2, 7, 9, 15, 17, 19, 22, 30, 39]. Our proposal is more scalable than past research which requires issuing *multiple* probes from landmark servers distributed around the world, to *each* individual target IP. We do not require multiple measurements targeted at each address from multiple vantage points. Whereas previous work often focuses on locating a handful of IP addresses in a few US universities [9, 17, 30, 39], our technique allows us to locate millions of addresses. Another advantage is that our results have high accuracy with median error down to 4.3 kilometers, while previous results showed error distance sometimes in the order of hundreds of kilometers [9, 30]. Note that, as described in Section 3, each data point in our ground truth set is shifted by 0.5 km in random direction to preserve privacy. Furthermore, since we do not need to perform complex multilateration, our computational requirements are more modest [9].

### 2.2 Network Topology

Network Topology geolocation methods combine network delay with information on network structure to achieve increased accuracy. *GeoCluster*, also proposed by Padmanabhan and Subramanian, combines BGP routing information with sparse IPs of known locations to assign geographic locations to whole address prefixes [30]. Jayant and Katz-Bassett extend CBG's ping-based approach by adding information from traceroutes [15]. They hypothesize that targets that follow similar traceroute paths also have similar delay-to-distance conversion characteristics and propose two approaches, Path-Based Estimation (*PBE*) and Router-Based Estimation (*RBE*). Katz-Bassett et al. later also propose *TBG*, which uses traceroute from landmark servers to the IP target and perform global optimization to find the location of both landmarks and targets [17]. More recently, Ciavarrini et al. presented a framework to understand how the position of landmarks and their distribution affect localization performance [2]. Multiple systems such as Octant [38], Alidade [1], or HLOC [34] combine delay measurement methods with other data sources such as reverse DNS and WHOIS information.

As with network delay approaches, typical network topology methods also suffer from low coverage and/or low accuracy. For instance, on a set of 265 target IPs located at universities in the United States, GeoCluster achieves a median error of 28 kilometers. This approach performs well with hosts located on university campuses, but it performs much worse on a larger and more realistic dataset of 181,246 IPs, with median error degrading to 685 kilometers [30]. The PBE and RBE approaches proposed by Jayant and Katz-Bassett achieve a median error of 376 kilometers and 346 kilometers, respectively [15]. The best variant of TBG, which also uses reverse DNS location hints based on the hostname of IP addresses, achieves a median error of 67 kilometers [17].

**In addition to being more accurate than previous network topology geolocation research, our method has other advantages** as we aim to extract all useful geolocation information embedded in traceroutes. First, in contrast to previous research, we do not specifically require the traceroutes to be directed to any particular targets. Instead, our approach can make use of datasets with sufficiently many random traceroutes. Second, we do not require traceroutes to the same IP address from multiple vantage points on the Internet. Instead it is enough to see the same pairs of IP addresses in the segments of multiple traceroutes, even if the source and target of the traceroutes are different. Third, from each traceroute instance we can extract information about multiple pairs along the path, not only between source and destination.

**The main disadvantage of our approach is that it requires a larger ground truth seed list of IPs with known location.** Traditional latency based approaches only require that the location of the landmarks be known. Our method needs a larger set of IPs with known location, on the order of millions of IPs. We use this seed list for location propagation.

### 2.3 IP Interpolation

Although it is used to augment other geolocation techniques and it is never used on its own, IP Interpolation is widely used in geolocation research to fill in the gaps of addresses with unknown location in IP ranges. Previously mentioned research projects GeoCluster [30, 31] and Alidade [1] use interpolation to increase coverage. Structon [10] is an approach proposed by Guo et al. that mines the contents of Chinese websites for mentions of locations, using regular expressions. The authors assign these locations to the IP addresses of the web servers hosting this content. They then use IP location interpolation to increase both accuracy and coverage by estimating the location of entire IP ranges from the location of few individual constituent IP addresses. In Checkin-Geo, Liu et al. use checkins logged by a location sharing social network for IP geolocation [23]. They also apply IP location interpolation to expand IP coverage. Lee et al. combine self-reported location data from a Korean crowd-sourced broadband speed test with IP location interpolation to assemble a detailed geolocation database [21]. To increase IP coverage, they perform interpolation using a majority rule vote with a threshold of 80% to assign individual IP address locations to entire IP ranges.

There are two aspects which are novel in our usage of IP interpolation. First, to our knowledge we use the largest ground truth dataset used to confirm the accuracy of IP interpolation. Second, we combine IP interpolation with traceroutes to propagate locations.

## 3 DATASETS AND PRIVACY

Online privacy is becoming increasingly important. Pew Research has found in 2016 that while many Americans are willing to share personal information in exchange for accessing online services, they are often cautious about disclosing their information and are frequently unhappy about what happens to that information once companies have collected it [33]. We have designed both our approach and our evaluation with this sensitive subject in mind.

We carried out our experiments on data from late 2017, since it was the time frame for which we could source all data sets at roughly the same time.

The public **traceroute dataset** contains 9 billion traceroutes collected between January and November 2017. We derived it from the *IPv4 Routed /24 Topology Dataset* [13] provided by the Center for Applied Internet Data Analysis (CAIDA). They collect this data through the Archipelago (Ark) Measurement Infrastructure, which spans approximately 208 servers located in 63 countries. Every 48 hours a random IP address is chosen in *each* /24 prefix, then the chosen IP addresses are individually probed by random Ark servers. Therefore, both the IP chosen per range and the Ark machine probing that IP change in time. While this allows for more data variety, it also prevents using the dataset for typical latency multilateration [9]. We further parse and apply post-processing on this dataset to extract latency neighbors, as described in Section 4.1. Since traceroutes are public and can be obtained from any Internet connected machine, this data has a low impact on user privacy.

Our proprietary **ground truth set** contains 8.9 million IP addresses with known location, compiled during the 28-day period ending on December 1st, 2017 from Bing query logs. It is one of the largest and most diverse ground truth sets used in geolocation literature. The dataset is derived from devices with global positioning sensors, where users opted-in to provide location information. It contains both mobile and fixed broadband IP addresses, since users often connect their mobile phones to their home Wi-Fi. The data covers the entire world. We described this type of proprietary dataset in more detail in our previous geolocation research [3–5]. Throughout this paper we used this ground truth set for both training and testing by performing **ten-fold cross validation**, by randomly splitting the data into ten folds and repeatedly using 9 folds for training and the last fold for testing. We report the results as the average of all the runs.

We never had access to the raw location data. Instead, the dataset was anonymized by an automated pipeline by aggregating all locations reported for an IP address, then adjusting the centroid of each IP address by 584 meters in a random direction. IP addresses with a large variance in reported locations were removed as outliers. These anonymized coordinates cannot be used to pinpoint individual addresses, but can locate an IP at a neighborhood level. While throughout this paper we refer to this location data as derived from *GPS* for succinctness, the dataset actually covers all global positioning systems, including *GPS*, *GLONASS*, *Galileo*, etc. [28]. Although we evaluate our approach on IPv4, methods described here can be equally applied to IPv6 IPs.

To aid in reproducing our experiments, we also perform our final evaluation with a second public training set extracted from **PeeringDB**, which is a self-reported database of worldwide peering points [24]. The dataset contains approximately 400 IP ranges spanning 128,000 IP addresses, along with geographic coordinates. Whereas the first ground truth set is proprietary and it mainly contains end user client IPs, this second dataset contains IP ranges that are part of Internet peering infrastructure. To obtain it, we enumerate the Internet Exchanges in the database, then for each exchange

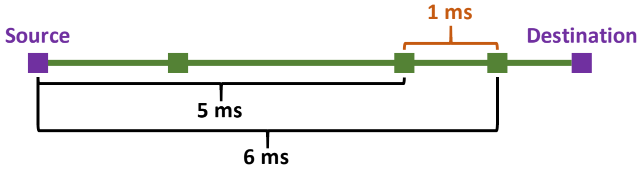

**Figure 1: Example of latency neighbors for $X <= 1$ ms. Since the latency difference between nodes 4 and 5 on the traceroute path is 1 milliseconds, we consider them latency neighbors.**

we enumerate the facilities. We retain only the facilities which contain exact location coordinates. Since IP ranges are published at the Internet Exchange level, we determine a location consensus among the coordinates of all the facilities belonging to an exchange. If all facilities are located in the same city, then we output the IP range and the consensus coordinates. Since this database is public and self-reported, the impact on user privacy is minimal. Along with this paper we are making the traceroute dataset parsing library [1] and the PeeringDB parsing and generation library [2] available as **open source**.

# 4 PRELIMINARY INVESTIGATION

We begin by testing our two assumptions. First, we define the concept of latency neighbors and we show that there is a relationship between neighbor latency and geographic distance. Second, we study whether addresses that are in the same IP range are also located in the same geographic region.

## 4.1 Latency neighbors

Traceroute tools reveal the path taken by packets that travel from one Internet connected device to another. They also measure the latency to each hop in the path [26]. The traceroute dataset we use contains the round-trip times (RTT) between source IPs and each reachable node on the path.

We define *latency neighbors* as pairs of nodes along a traceroute path that are within $X$ milliseconds from each other. We obtain the latency difference by subtracting the round-trip times between the source IP and the two neighbor candidates. Figure 1 shows an example of two nodes along a traceroute path which are at 5 and 6 milliseconds distance respectively from the source IP, which results in a latency difference of 1 millisecond. Since a pair can appear in multiple traceroutes, we aggregate all such instances and make a decision on the median round trip time. This aggregation has an added benefit of removing outlier pairs where the latency across multiple readings is too variable and the median becomes too high. To be considered latency neighbors, the nodes do not have to be located consecutively on the traceroute path, as long as they are within $X$ milliseconds of each other.

Our first assumption is that traceroute neighbors that are close together in terms of latency are also close geographically. As a preliminary test of this assumption, we extracted all latency neighbors from the traceroute dataset that were at most 10 milliseconds apart. Note all latencies are from round-trip measurements, so the actual latency between them was at most 5 milliseconds. Previous research has found that packets travel in real networks at about 4/9 the speed of light, or about 133 kilometers per millisecond [17]. We then further filtered these neighbor pairs to retain only the ones where both IPs were also present in our ground truth set with known IP locations. Since we required both exact neighbor IPs to be present in the ground truth set, this resulted in only 2,000 pairs, which is an extremely small coverage that makes it difficult to draw overall conclusions. The results show 65% of the neighbors are within 10 kilometers of each other. Although the results are promising, there is still a need to explore ways to increase ground truth coverage and to develop a systematic way to propagate locations over traceroute paths.

## 4.2 Colocation of IP Range Addresses

Our second assumption is that IP addresses that are in the same contiguous IP range are likely to also be located in the same geographic area. Although prior research has touched upon this observation [30], here we systematically test this hypothesis using our large ground truth set.

We segmented the IPv4 address space into IP ranges of varying lengths for netmasks between */28* (16 IPs) and */20* (4,096 IPs). We then extracted all pairs of IP addresses which are part of the same IP range, and are also in the ground truth set. So, for example, IPs 50.121.73.3 and 50.121.73.47 would form a pair in IP ranges with netmask */26* (64 IPs) to */20* (4,096 IPs) or larger, but not for IP range *50.121.73.0/28* which is only 16 IPs in size. Since we know the location of each IP in the ground truth set, we were then able to compute the geographic distance between the IPs in each pair.

Figure 2 presents the results as cumulative distance curves. The X axis represents the distance between pair items and the Y axis shows how many pairs are within that distance. For instance, if we look at the first column (*<10 km*) for the IP range size of 1,024 IPs, we can observe that roughly 60% of pairs in this type of IP range are within 10 kilometers of each other. We can draw two conclusions from the graph. First, the size of the IP range is directly proportional to pair distance. As the range size increases, the distance between pairs in the range also increases. For ranges that are 256 IPs in size, 88% of pairs are within 30 kilometers of each other, but that percentage drops to 77.9% for IP ranges with 1,024 IPs. Second, even if these preliminary results are promising, the graph shows that there is room for improvement. We cannot assign the locations of the pairs to the entire IP range, since in some cases the locations are contradictory. We must therefore find a way to further filter these IP ranges to retain the ones where the location signal is consistent.

# 5 IP RANGE INTERPOLATION

We apply our finding that addresses in the same IP range are often colocated geographically to the problem of increasing ground truth IP coverage. We propose performing *IP range interpolation* by finding IP ranges where all ground truth IPs contained in that range are in the same geographic region, and then assigning the center of all IP coordinates to the entire range. Figure 3 shows an example where an IP range contains two ground truth IPs, both located in New York City. Since all the ground truth IPs contained in this IP

[1] https://github.com/zmarty/ScamperTracerouteParser
[2] https://github.com/zmarty/PeeringDBToTSV

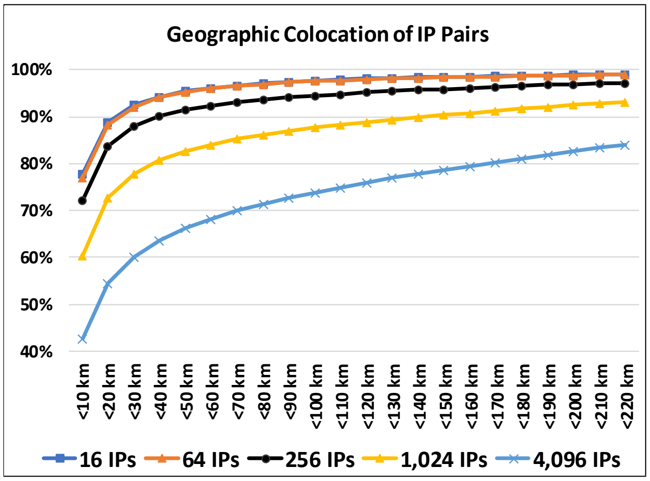

**Figure 2: Cumulative distance between pairs of IPs in the same range, for different range sizes. For example, in the *<20 km* column, the value of the distance for IP ranges of *256 IPs* is 83.8%. This means that if we segment the IP space in contiguous ranges of size 256, a pair of IP addresses from the ground truth set that are in the same IP range are at a distance smaller than 20 kilometers from each other in 83.8% of cases.**

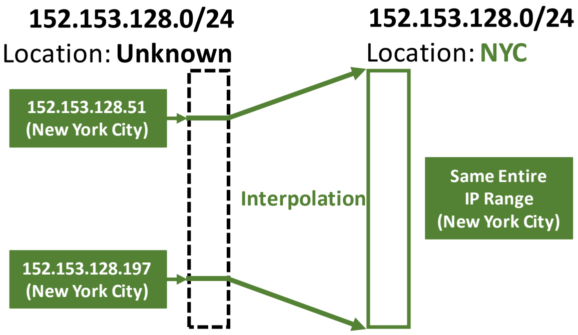

**Figure 3: Example of IP range interpolation. Since IP range 152.153.128.0/24 contains two IP addresses with the same known location (coordinates in New York City), we propagate that location to the entire IP range.**

range are located in the same region, we propose assigning the location New York City to the entire IP range.

To perform interpolation, for a given IP range size we first grouped all ground truth IPs by the given netmask. In each of these contiguous IP ranges we computed the pairwise distance between the ground truth IPs. We retained the IP ranges that contain at least *n* ground truth IPs in total, and *all* these IPs in the range are within *m* kilometers of each other. We then assigned the center of all these coordinates to be the location of the entire IP range. We evaluated our proposal on multiple IP range sizes and multiple values of *n* and *m* using local parameter search on our ground truth set. We found that using a size of 256 IPs yields the best combination of accuracy and coverage. This IP range size is also typically used by commercial IP geolocation databases. We obtained good accuracy by setting the *n* parameter to 2. Setting it to 1 yields lower accuracy by 0.6 percentage points and higher coverage by 0.8%, while setting it to 3 yields higher accuracy by 0.2 percentage points and a decrease in coverage by 0.8%. Finally, we set the *m* parameter to 25 kilometers. Setting it to 20 kilometers yields a 1 percentage point improvement in accuracy at 10 kilometers, at the cost of reduced coverage by 25%, while setting it at 30 kilometers results in an accuracy decrease of 1.5 percentage points but with an increase of 11% in coverage. Using these parameters (*n=2, m=25*), we filtered the 3.5 million distinct IP ranges of size 256 in the ground truth set down to 1.5 million ranges used for interpolation.

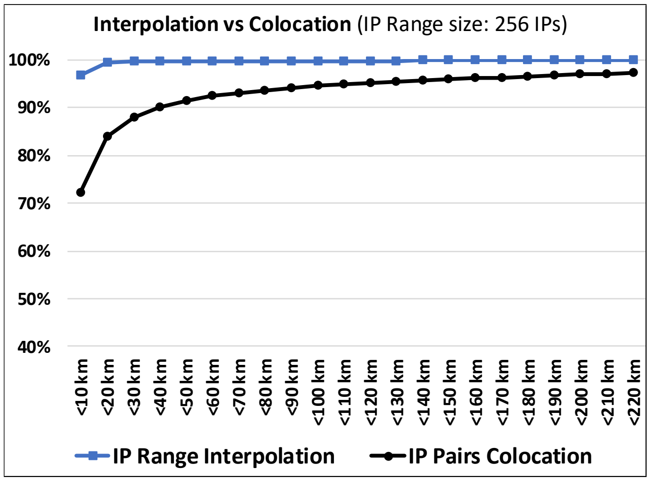

**Figure 4: Comparison of IP range interpolation error distance to IP range colocation distance. Note that interpolation error distance represents the distance between the predicted location and the actual location as given by the ground truth set, while the colocation distance represents the pair-wise distance between IPs with known location.**

Figure 4 presents the evaluation result using ten-fold cross validation on the ground truth set. The interpolation curve shows the *error distance* between where our interpolation places the IP, and its actual location. Results show that 96.7% of IPs have a predicted location that is within 10 kilometers of their actual location, and 99.4% of them are within 20 kilometers. For comparison, we have also displayed the equivalent 256 IPs colocation curve from Figure 2, where IP range pairs were located within 10 kilometers for 72.3% of data points, and 20 kilometers for 83.8% of data points. In conclusion, if two or more IP addresses in the same IP range are located in the same geographic area, it is very likely that the rest of the IP range is also in the same area. Using IP range interpolation we effectively increased our ground truth coverage from 8.9 million IP addresses to 382 million IP addresses.

## 6 LOCATION PROPAGATION EVALUATION

We perform location propagation by combining the concepts of latency neighbors and IP interpolation. First, we interpolate the ground truth set to increase its coverage. Then, we propagate these locations from IPs with known location to IPs with unknown location, through latency neighbors. Figure 5 presents a simplified example of how locations are propagated. We use the interpolated ground truth set both for training and testing by using ten-fold

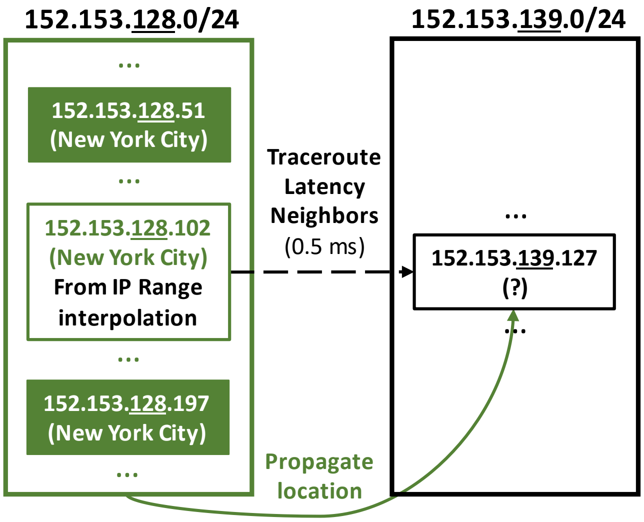

**Figure 5: Example of location propagation along the traceroute path. Since the two IPs are latency neighbors, we can propagate the location of the IP on the left to the IP on the right.**

cross validation, where we propagate locations using 9 folds and we test using the last one. We report the results as the average of all the runs. The balance of accuracy and coverage of the model can be adjusted by varying two parameters used in determining latency neighbors: *X*, the maximum latency difference between two nodes and *Y*, a new parameter which restricts the maximum RTT between the source IP, and any of the two neighbors. In Figure 1 the maximum latency difference between the source IP and any of the two candidate latency neighbors is 6 milliseconds. If we set *Y* to be 6 or larger, then the two neighbors would be extracted as valid.

To find the optimal values we performed a local parameter search. We found that as we increased the maximum latency difference *X* from 1 to 4 the accuracy at *<10 km* decreased and the coverage increased. The same was true for varying the maximum RTT parameter from 1 to 10. The curves *Traceroute-GPS-HighAcc* and *Traceroute-GPS-HighCov* in Figure 6 present two instances of these parameters that graphically demonstrate the effect on accuracy. The former variation plots the results for parameters *X=2, Y=2* and the latter variation uses *X=3, Y=9*. The higher accuracy version has a coverage of 1.4 million IP addresses across 7,400 IP ranges with propagated location, while the higher coverage version has a coverage of 15 million IPs across 83,000 IP ranges. These IPs had a previously unknown location that we now determined using location propagation. This evaluation uses the interpolated GPS-based ground truth as both training and test set. We also ran the same experiments on the non-interpolated ground truth set and obtained very similar results, but at lower IP coverage. Figure 6 also shows the results for a third variation of our approach that uses the PeeringDB dataset with *X=3, Y=9* for location propagation. Here the overall results roughly fall between the first two instances.

We compare these three variations against two state of the art commercial databases, one labeled *ProviderA* and the other labeled *ProviderB*. We cannot reveal the names of the proprietary databases since their terms of use forbid comparative benchmarking. The results show that all three variants of our approach consistently outperform the commercial databases in error distance. Table 1 also compares our variants to these two baselines across multiple metrics. Our three instances outperform the commercial databases both in terms of median error (lower is better) and percentage of data points with error <10 km (higher is better). The last column of the table displays root-mean-square error, which is a metric more heavily influenced by outliers. It shows that the two instances trained on the proprietary dataset have a better (lower) RMSE than the commercial providers. However, it shows that the instance derived from PeeringDB data contains some outliers with high error distance. One potential explanation for these outliers is that sometimes PeeringDB IP ranges contain IPs that interconnect datacenters that are far from each other, so errors caused by these IP ranges result in large error distances.

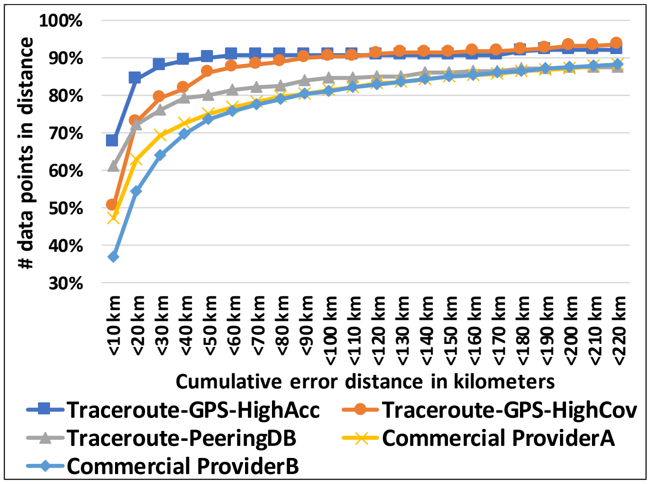

**Figure 6: Cumulative error distance results that compare three instances of our approach to two state-of-the-art commercial IP geolocation baselines.**

**Table 1: Comparison between three instances of our two approaches and two state of the art commercial geolocation databases, across several metrics. As described in Section 3, each data point in our ground truth set is shifted by 0.5 km in random direction to preserve privacy.**

| | Median error | % Err <10km | RMSE in km |
|---|---|---|---|
| **Traceroute-GPS-HighAcc** | 4.3 km | 67.7% | 329.3 |
| **Traceroute-GPS-HighCov** | 10.1 km | 50.5% | 423.6 |
| **Traceroute-PeeringDB** | 8.4 km | 61.1% | 2124.9 |
| **Commercial Provider A** | 11.1 km | 47.2% | 545.9 |
| **Commercial Provider B** | 16.7 km | 36.7% | 545.3 |

## 7 CONCLUSIONS AND FUTURE WORK

We investigated and combined two IP geolocation approaches, one based on IP range interpolation, and the other one based on location propagation over traceroute paths. Our combined technique significantly outperforms state of the art commercial databases by up to 31 percentage points at error distance smaller than 10 kilometers.

To aid in reproducing our results, we are making the traceroute dataset parsing library, and the PeeringDB parsing and generation library available as open source. One potential area for future work would be to exploit more information available in traceroute paths. In previous work we have shown that reverse DNS hostnames can be a good source of geolocation information [4, 5]. In addition to using locations from ground truth addresses, it should therefore be possible to parse the reverse DNS hostnames of nodes along traceroute paths to extract and propagate more location hints.

## REFERENCES

[1] Balakrishnan Chandrasekaran, Mingru Bai, Michael Schoenfield, Arthur Berger, Nicole Caruso, George Economou, Stephen Gilliss, Bruce Maggs, Kyle Moses, David Duff, et al. 2015. Alidade: Ip geolocation without active probing. *Department of Computer Science, Duke University, Technical Report* (2015).

[2] Gloria Ciavarrini, Maria S Greco, and Alessio Vecchio. 2018. Geolocation of Internet hosts: Accuracy limits through Cramér–Rao lower bound. *Computer Networks* 135 (2018), 70–80.

[3] Ovidiu Dan, Vaibhav Parikh, and Brian D Davison. 2016. Improving IP geolocation using query logs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 347–356.

[4] Ovidiu Dan, Vaibhav Parikh, and Brian D Davison. 2018. Distributed Reverse DNS Geolocation. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 1581–1586.

[5] Ovidiu Dan, Vaibhav Parikh, and Brian D Davison. 2018. IP geolocation through reverse DNS. *arXiv preprint arXiv:1811.04288* (2018).

[6] Ziqian Dong, Rohan DW Perera, Rajarathnam Chandramouli, and KP Subbalakshmi. 2012. Network measurement based modeling and optimization for IP geolocation. *Computer Networks* 56, 1 (2012), 85–98.

[7] Brian Eriksson, Paul Barford, Joel Sommers, and Robert Nowak. 2010. A learning-based approach for IP geolocation. In *International Conference on Passive and Active Network Measurement*. Springer, 171–180.

[8] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, 463–469.

[9] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. 2006. Constraint-Based Geolocation of Internet Hosts. *IEEE/ACM Transactions on Networking* 14, 6 (Dec 2006), 1219–1232. https://doi.org/10.1109/TNET.2006.886332

[10] Chuanxiong Guo, Yunxin Liu, Wenchao Shen, H.J. Wang, Qing Yu, and Yongguang Zhang. 2009. Mining the Web and the Internet for Accurate IP Address Geolocations. In *INFOCOM 2009*. 2841–2845. https://doi.org/10.1109/INFCOM.2009.5062243

[11] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 527–538.

[12] Cheng Huang, D.A. Maltz, Jin Li, and Albert Greenberg. 2011. Public DNS system and Global Traffic Management. In *INFOCOM 2011*. 2615–2623. https://doi.org/10.1109/INFCOM.2011.5935088

[13] Young Hyun, Bradley Huffaker, Dan Andersen, Emile Aben, Colleen Shannon, Matthew Luckie, and K Claffy. [n.d.]. The CAIDA IPv4 routed/24 topology dataset. http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml. Accessed: 2019-01-02.

[14] IP2Location.com. 2018. Geolocate IP Address Location using IP2Location. https://www.ip2location.com/ Accessed: 2018-08-13.

[15] Rika Jayant and Ethan Katz-Bassett. 2004. Toward Better Geolocation: Improving Internet Distance Estimates Using Route Traces. *Report, The Pennsylvania State University* (2004).

[16] Hao Jiang, Yaoqing Liu, and Jeanna N Matthews. 2016. IP geolocation estimation using neural networks with stable landmarks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*. IEEE, 170–175.

[17] 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*. ACM, 71–84.

[18] Bernhard Kölmel and Spiros Alexakis. 2002. Location Based Advertising. In *First International Conference on Mobile Business*. Athens, Greece.

[19] Sándor Laki, Péter Mátray, Péter Hága, István Csabai, and Gábor Vattay. 2010. A model based approach for improving router geolocation. *Computer Networks* 54, 9 (2010), 1490–1501.

[20] Sándor Laki, Péter Mátray, Péter Hága, Tamás Sebők, István Csabai, and Gábor Vattay. 2011. Spotter: A model based active geolocation service. In *INFOCOM, 2011 Proceedings IEEE*. IEEE, 3173–3181.

[21] Yeonhee Lee, Heasook Park, and Youngseok Lee. 2016. IP Geolocation with a crowd-sourcing broadband performance tool. *ACM SIGCOMM Computer Communication Review* 46, 1 (2016), 12–20.

[22] Dan Li, Jiong Chen, Chuanxiong Guo, Yunxin Liu, Jinyu Zhang, Zhili Zhang, and Yongguang Zhang. 2012. IP-Geolocation mapping for moderately-connected Internet regions. *IEEE Transactions on Parallel and Distributed Systems* (2012).

[23] Hao Liu, Yaoxue Zhang, Yuezhi Zhou, Di Zhang, Xiaoming Fu, and KK Ramakrishnan. 2014. Mining checkins from location-sharing services for client-independent ip geolocation. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 619–627.

[24] Aemen Lodhi, Natalie Larson, Amogh Dhamdhere, Constantine Dovrolis, et al. 2014. Using peeringDB to understand the peering ecosystem. *ACM SIGCOMM Computer Communication Review* 44, 2 (2014), 20–27.

[25] Lori MacVittie. 2012. Geolocation and Application Delivery. https://www.f5.com/pdf/white-papers/geolocation-wp.pdf. Accessed: 2018-08-02.

[26] Gary Scott Malkin. 1993. Traceroute using an IP option. Internet Requests for Comments. 2317 (January 1993). https://tools.ietf.org/html/rfc1393

[27] MaxMind, Inc. 2018. Detect Online Fraud and Locate Online Visitors. https://www.maxmind.com/en/home Accessed: 2018-08-13.

[28] Pratap Misra and Per Enge. 2006. Global Positioning System: signals, measurements and performance second edition. *Massachusetts: Ganga-Jamuna Press* (2006).

[29] Neustar, Inc. 2018. IP Intelligence. https://www.security.neustar/digital-performance/ip-intelligence Accessed: 2018-08-13.

[30] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. 2001. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *SIGCOMM 2001* (San Diego, California, USA). ACM, San Diego, California, USA, 173–185. https://doi.org/10.1145/383059.383073

[31] Venkata N Padmanabhan and Lakshminarayanan Subramanian. 2001. Determining the geographic location of Internet hosts. In *SIGMETRICS/Performance*. 324–325.

[32] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review* 41, 2 (2011), 53–56.

[33] Lee Rainie and Maeve Duggan. 2016. Privacy and information sharing. *Pew Research Center* 16 (2016).

[34] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. 2017. HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks. *arXiv preprint arXiv:1706.09331* (2017).

[35] Yuval Shavitt and Noa Zilberman. 2011. A geolocation databases study. *IEEE Journal on Selected Areas in Communications* 29, 10 (2011), 2044–2056.

[36] Craig A. Shue, Nathanael Paul, and Curtis R. Taylor. 2013. From an IP Address to a Street Address: Using Wireless Signals to Locate a Target. In *WOOT 2013* (Washington, D.C.). USENIX, Washington, D.C. https://www.usenix.org/conference/woot13/workshop-program/presentation/Shue

[37] Dan Jerker B Svantesson. 2007. E-Commerce Tax: How The Taxman Brought Geography To The 'Borderless' Internet. *Revenue Law Journal* 17, 1 (2007), 11.

[38] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. 2007. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *NSDI 2007* (Cambridge, MA). USENIX Association, Berkeley, CA, USA, 23–23. http://dl.acm.org/citation.cfm?id=1973430.1973453

[39] Inja Youn, Brian L. Mark, and Dana Richards. 2009. Statistical Geolocation of Internet Hosts. In *Proceedings of 18th International Conference on Computer Communications and Networks (ICCCN)*. 1–6. https://doi.org/10.1109/ICCCN.2009.5235373