Networking 2013 1569696825

Improving IP Geolocation by Crawling the Internet PoP Level Graph

Yuval Shavitt
School of Electrical Engineering
Tel-Aviv University
Tel Aviv, Israel
Email: shavitt@eng.tau.ac.il

Noa Zilberman
School of Electrical Engineering
Tel-Aviv University
Tel Aviv, Israel
Email: noa@eng.tau.ac.il

***Abstract*—The geographical location of Internet IP addresses is important for academic research, commercial and homeland security applications. While commercial databases claim to have a very high level of accuracy, the correctness of their databases is questionable. Academic tools, based on delay measurements, were shown to have a large range of error. We present a novel algorithm that crawls the Internet PoP level graph to improve the accuracy of geolocation, combining information from both geolocation databases and delay measurements. We show that the results provided by the algorithm are more accurate than geolocation databases information while avoiding the pitfalls of delay measurements' usage.**

I. Introduction

Geolocation services have become more and more of a necessity in many fields and for many applications. While the end user is not always aware of it, many websites visited every day use geolocation information for targeted localized advertising, localized content (such as local news and weather), and compliance with local law.

One of the most highlighted purposes of geolocation information is for fraud prevention and as a mean of security. Banking, trading, and almost any other type of business that handles online money transactions are exposed to phishing attempts as well as other schemes. Criminals try to break into user accounts to transfer money, manipulate stocks, make purchases and other illegal activities. Geolocation information provides means to reduce the risk, for example by blocking users from certain high-risk countries and cross-referencing user expected and actual locations. This type of IP geolocation is many times coarse, and does not need to geolocate the user on the street level. Geolocation information is also important in many research fields. It improves Internet mapping and characterization, as it ties the Internet graph to actual node positions, and allows exploring new aspects of the network that are otherwise uncovered, such as the effect of ISP location on its services and types of relationships with other service providers.

Geolocation services range from free services to ones that cost tens of thousands of dollars a year. Since assigning a location to an IP address is a complicated task, services use a variety of methods. The most basic geolocation services use DNS resolution as the basis for the database [28], while others use proprietary means such as random forest classifier rules, hand-labeled hostnames [1], user's information provided by partners [3] and more.

One group of geolocation mapping services is based on network measurements. IP2Geo [22] was one of the first to suggest a measurement-based approach to approximate the geographical distance of network hosts. A more mature approach is constraint based geolocation [11], using several delay constraints to infer the location of a network host by a triangulation-like method. Later works, such as Octant [30], used a geometric approach to localize nodes within a 22 miles radius. Katz-Bassett *et al.* [15] suggested topology based geolocation using link delay to improve the location of nodes. Leveraging buffering delay estimation at each hop for geolocation purposes was proposed by Gueye *et al.*[10]. Yoshida *et al.* [31] used end-to-end communication delay measurements to infer PoP level topology between thirteen cities in Japan. Eriksson *et al.* [5] applied a learning based approach to improve geolocation. They reduced IP geolocation to a machine learning classification problem and used Naive Bayes framework to increase geolocation accuracy. One online geolocation service that allows querying specific IP addresses is Spotter, which is based on a work by Laki *et al.* [17]. Spotter uses a probabilistic geolocation approach, which is based on a statistical analysis of the relationship between network delay and geographic distance.

Only a few works have focused on the accuracy of geolocation databases, all showed them to be inaccurate: In 2008, Siwpersad *et al.* [27] examined the accuracy of Maxmind [19] and IP2Location [12]. They assessed their resolution and confidence area and concluded that their resolution is too coarse and that active measurements provide a more accurate alternative. Gueye *et al.* [9] investigated the imprecision of relying on the location of blocks of IP addresses to locate Internet hosts and concluded that geolocation information coming from exhaustive tabulation may contain an implicit imprecision. Muir and Oorschot [20] conducted a survey of geolocation techniques used by geolocation databases and examined means for evasion/circumvention from a security standpoint. In a previous work [25], we studied extensively seven geolocation services both on the IP and PoP level. We showed that the information in the databases may be largely biased at the ISP level. Additionally, correlation was found between databases: some databases, such as Maxmind [19] and IP2Location [12] have an extremely small median distance between an IP address' geolocations, below 10km, while for other databases, such as GeoBytes [7] and HostIP.info [13] the median distance may be above 500km. The differences between databases is sometimes even in the range of countries. Poese *et al.* [23] studied five databases and showed that while on the country-level they are rather accurate, the databases are highly biased towards a few popular countries.

Constraint based approaches are many times no better than geolocation databases. They inherently have an inaccuracy in the range of tens to hundreds of kilometers [25], and strongly depend on the location of the vantage points. A non optimal location of vantage points may lead to an error in the range of hundreds of kilometers, and more.

In this paper we propose a novel algorithm for improving IP geolocation, using PoP level maps. The algorithm, operating at the PoP level, combines information from geolocation databases with PoP-level link delays to clean anomalies and false information and increase the accuracy of geolocation.

II. PoP Level Maps

Service providers tend to place multiple routers in a single location called a Point of Presence (PoP), which serves a certain geographical area. A PoP is defined as a group of routers which belong to the same AS and are physically located at the same building or campus. The PoP extraction algorithm used in this work [6] is based on the fact that in most cases [8], [24] the PoP consists of two or more backbone/core routers and a number of client/access routers. The client/access routers are connected redundantly to more than one core router, while core routers are connected to the core network of the ISP. The algorithm takes a structural approach and looks for bi-partite subgraphs* with certain weight constraints in the IP interface graph of an AS; no aliasing to routers is needed. The bi-partite subgraphs serve as the cores of the PoPs and are extended with other nearby interfaces.

The algorithm works on the Interface graph of each ISP separately. It starts by removing all edges with delay higher than PD_{max_th}, PoP maximal diameter threshold, and edges with number of measurements below PM_{min_th}, the PoP measurements threshold. As a result the ISP interface graph is partitioned to several components, each is a candidate to become one or more PoPs. Next, the algorithm looks at the bi-partites in each component and uses the rich connectivity there between the sources (parents) and destinations (children) to check for node co-location based on link delays between the groups. If *parent pair* and *child pair* groups are connected, then the weighted distance between the groups is calculated (If they are connected, by definition more than one edge connects the two groups); if it is smaller than a certain threshold the pair of groups is declared as part of the same PoP. Last, a unification of loosely connected parts of the PoP is conducted.

*A bipartite graph is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V

For this end, the algorithm looks for connected components (PoP candidates) that are connected by links whose median distance is very short (below PD_{max_th}).

The basic PoPs geolocation algorithm, which is referred to as the **naive geolocation algorithm**, is based on geolocation databases: it uses the geographic location of each of the IPs included in a PoP, as denoted by at least three geolocation databases (typically more) and takes the median location. A range of error, indicating the radius within 50% of the IP location votes reside, is assigned per PoP and the location of a PoP is further refined based on these locations alone. As all the PoP IP addresses should be located within the same campus, the location confidence of a PoP is significantly higher than the confidence that can be gained from locating each of its IP addresses separately. A detailed discussion of these algorithms appears in [6], [25].

The connectivity between PoPs is an important part of PoP level maps. DIMES generates PoPs connectivity graph using unidirectional links. We define a link L_{SD} as the aggregation of all unidirectional edges originating from an IP address included in a PoP S and arriving at an IP address included in a PoP D. Each of the IP level links has an estimate of the median delay measured along it, with the median calculated on the minimal delay of up to four consecutive measurements. Every such four measurements comprise a basic DIMES operation. All measured values are roundtrip delays [6]. A Link has the following properties:

- Source and Destination PoP nodes.
- The number of edges aggregated within the link.
- Minimal and Maximal median delays of all IP edges that are part of the PoP level link.
- Mean and standard deviation of all edges median delays.
- Weighted delay of all edges median delays. The edge's weight is the number of times it was measured.
- The geographical distance between source and destination PoP, calculated based on the PoPs geolocation.

A weighted delay of a link is used to mitigate the effect of an edge with a single measurement on the overall link delay estimation, where a link is otherwise measured tens of times through other edges.

III. Crawling PoP Level Maps for Improved Geolocation

The PoP geolocation algorithm was found to work well [6], however it is not error free as it depends on the quality of the geolocation databases it uses. When the differences between databases are extreme, it fails to locate the related PoPs. We thus propose a method to improve this initial geolocation using a crawling algorithm. This method can be further expended to improve IP-level geolocation.

A property of our initial geolocation algorithm is that it gives a confidence to the PoP's location. Each PoP is assigned a range of error, being the minimal radius covering 50% or more of the PoP's IP addresses locations (but no more than 100 kilometers, a threshold set in the algorithm), and the confidence is derived from the percentage of IP-level locations included within this radius. Using PoPs with a known location (such as universities) and PoPs marked with a high level of confidence, we find the location of PoPs with a lower level of confidence.

The algorithm starts by identifying and marking the PoPs for which the location is certain. The algorithm then discovers PoPs that are located in the same place as the marked PoPs, based on a PoP-level link delay. Next it examines all the PoP's IP-level locations in the geolocation services, and finds one that gives the best delay-distance matching to marked neighboring PoPs. If no location passes a goodness threshold, multilateration from the marked neighboring PoPs is used. The algorithm then iterates and attempts to improve the location of PoPs that were not handled yet using the location of newly marked PoPs. Figure 1 shows the stages of the algorithm, detailed as follows:

Primary Anchor PoPs Marking Mark all PoPs with a definite known location as *anchors* (dark nodes), the rest of the PoPs (light nodes) are placed based on the previous naive geolocation algorithm [6]. Anchor PoPs belong to universities, research facilities, and other known locations.

Additional Anchor PoPs Marking Mark all PoPs with a high level of confidence as *anchors* . An anchor PoP can be used to geolocate other PoPs with a lower level of confidence. For high level of confidence the following three conditions are required:

- $P_{tot} \geq P_{tot\_th}$, where P_{tot} is the percentage of IP level locations within the PoP's error range and P_{tot_th} is a threshold for this parameter.
- $P_{IP} \geq P_{IP\_th}$, where P_{IP} is the percentage of IP level locations located within the PoP's location error range when "no location" replies are excluded and P_{IP_th} is a threshold for this parameter.
- $R \leq R_{th}$, where R is the location error range of the PoP and R_{th} is the range radius threshold for this parameter.

The anchor PoPs $\{B, F, I, N\}$, marked during the primary and additional marking stages, are shown as dark nodes in Figure 1a.

Co-Locate PoPs For each unmarked PoP node with a link delay below a certain threshold (D_{co_th}, typically less than $1ms$) to a marked PoP, one can assume that both PoPs are located in the same place. We thus define the PoPs as *co-located*, assign to the unmarked PoP the same location as the marked PoP, and add it to the group of marked PoPs. In Figure 1b the co-located PoPs are A,C,D,O since the link delays on edges (A,B), (B,C), (D,F), and (O,N) are all less than $D_{co\_th} = 1ms$ (1mS was selected for demonstration). After updating the geolocation of PoPs A,C,D,O they are marked.

Location Update by Delay and Geolocation Data For each unmarked PoP in the map with at least one neighboring marked PoP, POP_{M_i} $1 \leq i \leq k$ (k is the number of such neighbors), go over locations L_{IP} of all the IP addresses included in the unmarked PoP, as indicated by each geolocation service. For every L_{IP}, calculate the delay to distance ratio RA from the location L_{IP} to every POP_{M_i} and its corresponding link delay. If there is L_{IP} such that $RA_{min} \leq RA \leq RA_{max}$, set the location of the PoP to L_{IP} and mark it. If there is more than one such location select the location with the best lexicographic ordered RA ratio value. In our example (Figure 1c), PoP M has two neighboring marked PoPs: N and D. The delay from M to N is $7ms$ and from M to D - $10ms$. If we set $RA_{min} = 95km/ms$, $RA_{max} = 110km/ms$, we expect PoP M to be in the range of 665km to 770km from PoP N and 950km to 1100km from PoP D, as indicated by the shaded circles in the figure. While the location of PoP M was initially set by the majority vote of all geolocation databases, two more alternative locations were indicated by some of the databases: M1 and M2. Since M2 is located within the expected range from N and D it is selected as the location of PoP M, and the PoP is marked.

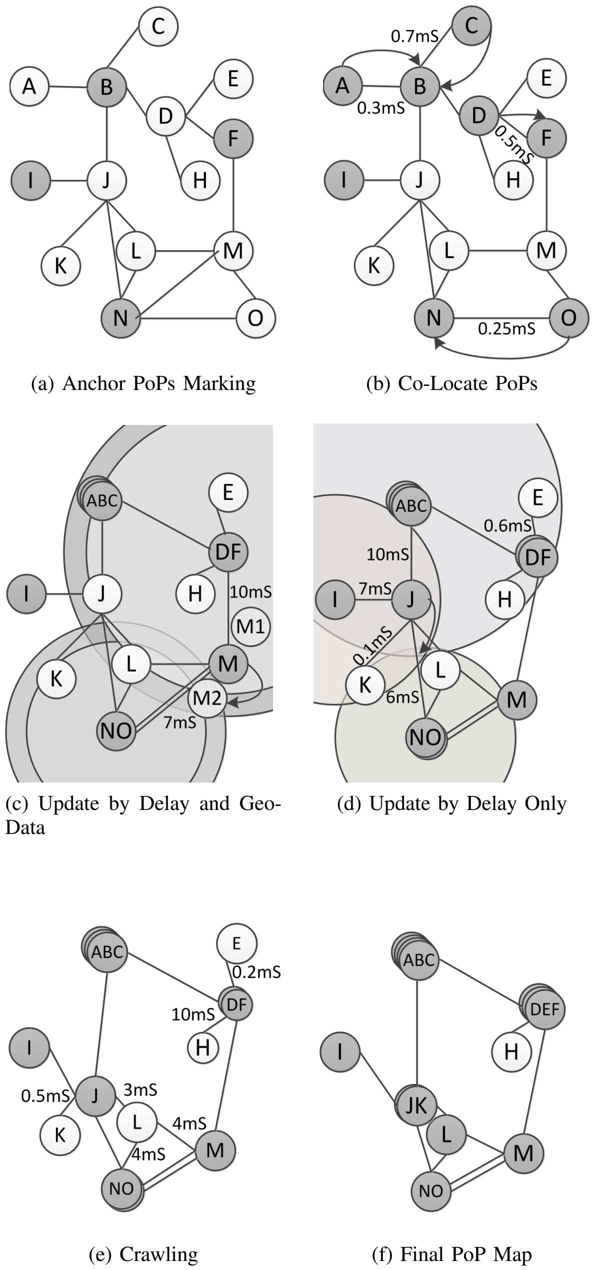
Location Update by Delay Only For each unmarked PoP node with at least three neighboring marked PoPs, POP_{M_i} $1 \leq i \leq k$ (k is the number of such neighbors), update the PoP location such that the ratio RA between the PoP's geographic distance from every POP_{M_i} and its corresponding link delay will be closest to the optimal ratio value RA_{opt}. In other words, multilaterate the PoP's location based on the delay from the marked PoPs and their geographic location and mark it. A constraints based approach is currently used for the multilateration, but other methods of multilateration may be used. In Figure 1d example, only node J is a candidate for location update by delay. The three shaded circles around PoPs B,I,N indicate the expected location of PoP J relative to each one of them. The location of PoP J is thus updated to the crossing area of all three ranges and the PoP is marked.

Crawling Iterate the Co-Locate and Location Update stages, using previously marked PoPs to update the location of non-marked ones. As a result, the PoPs locations are propagated through the PoPs network, such that PoPs with a high level of accuracy update the location of PoPs with a medium level of accuracy, and those in turn update others. The process ends after no PoP can be added to the group of marked PoPs. Figure 1e shows the map of geolocated PoPs after the first iteration: there are ten marked PoPs and four unmarked PoPs. PoPs E and K will be relocated in the second iteration, as their neighboring PoP is now marked and their link delay is less than 1ms. PoP L will be relocated in the Location Update by Delay Only step. PoP H either will be relocated in the Location Update by Delay and Geolocation Data step or will not be marked. The third iteration will have no updates and thus the crawling algorithm will terminate, as shown in Figure 1f.

The algorithm can be further extended to IP-level geolocation. For a given target IP address, take the following steps:

PoP Located If the target is part of a PoP, assign it the location of its hosting PoP.

CIDR/24 based If there is an IP address in a PoP with the same CIDR/24 as the target, assign to the target the location of the PoP. If multiple such IP addresses exists, use the location of the longest prefix match IP. We note that some loss of accuracy exists in such a case, but this provides at least the same level of accuracy as most geolocation databases.



(a) Anchor PoPs Marking (b) Co-Locate PoPs

(c) Update by Delay and Geo-Data (d) Update by Delay Only

(e) Crawling (f) Final PoP Map

Fig. 1: Crawling Algorithm

One-Hop Location If the target is one hop away from an IP in a PoP or an IP conforming with the CIDR/24 rule, and the edge delay is less than D_{co_th}, assign the target the same location as its one-hop neighbor.

Two-Hop Location same as above but with two hops.

PoP-IP Multilateration Find the three IP addresses which are part of different PoPs, with minimal delay from the target, and use multilateration for the target location.

The algorithm description above provides only a rough outline of the full Crawling algorithm due to space limitation. For example, the selection of marked PoPs is refined in a manner that excludes marked PoPs with an initial distance to delay ratio significantly different from other neighbor marked PoPs.

IV. Datasets

Two datasets are used in this study: one from 2012, and one from 2010, which was selected as it was carefully studied in our previous works [25], [6]. Both datasets use measurements from DIMES [4] and iPlane [18]. We note that the traceroute measurements are performed differently by DIMES and iPlane, as every DIMES measurement is combined of a train of four traceroute measurements, and only the best time of every hop is used for an edge delay calculation. This affects the results beyond a ratio of $1:4$ in the number of measurements. For example, we filter out faulty traceroute hops, such as IP and AS level loops on edge level. Over 170 million measurements are filtered out of the iPlane measurements, while only $61K$ such measurements are filtered from the DIMES data (DIMES filters some of the measurements before adding them to the database). Due to the differences, edges discovered by DIMES are annotated with delay information measured only by DIMES, and iPlane data is used to add edges that were not discovered by DIMES. iPlane typically increases the number of discovered edges by $\tilde{2}0\%$, but it measures only a small subset of the edges that DIMES discovered.

2010 Dataset The dataset is comprised of 478 million traceroutes conducted in weeks 42 and 43 of 2010, measured by 1308 DIMES agents and 242 iPlane vantage nodes. Five geolocation databases are used for the naive geolocation of the PoPs: MaxMind GeoIP City, IPligence Max, IP2Location DB5, GeoBytes and HostIP.info. The generated PoP level map contains 4750 PoPs and $87.3K$ IP addresses in 1697 different ASes. 4098 PoPs are discovered using the DIMES data alone.

2012 Dataset The measurements in this dataset are taken from weeks 19 and 20 of 2012. 203 million traceroutes were collected from 988 DIMES agents and 153 iPlane vantage points. Five geolocation databases are used for the naive geolocation of the PoPs: MaxMind GeoIPLite City, IPligence Max, HostIP.info, DB-IP and NeuStar's IP Intelligence (formerly Quova). The generated PoP level map contains 5215 PoPs and 98650 IP addresses in 2636 different ASes. This map contains also universities, research institutes and exchanges points, as in the 2010 dataset.

V. Results

1) Basic Results: The PoP Crawling algorithm is initially executed using very conservative thresholds, described in subsection V-2. The sensitivity of the algorithm to these parameters is studied later in this section.

Running the algorithm converges fast for both datasets: only five crawling iterations, the last iteration without any update. 47.3% of the 2010 PoPs and 34.7% of the 2012 PoPs are marked on the first iteration. The crawling algorithm results are broken in Table I by the algorithm stages (or relocation method). Less than 8% of the PoPs were relocated based on link delay only.

Between a quarter to a third of the PoPs are not affected by the crawling algorithm, and maintain their naive original

Crawling Algorithm Stage	Relocated PoPs	
	2010	2012
Anchors	17.9%	29.3%
Co-Location	28.3%	20.1%
Delay and Geolocation Data	21.3%	13%
Delay Only	7.6%	6.1%
Not Relocated	24.8%	31.5%

TABLE I: PoP Relocation By Algorithm's Stage

position. There are several reasons for not marking a PoP: First, the PoP may not be connected to any other PoP, which is the case for over a quarter of the unmarked PoPs in 2012. Note that such a PoP is connected to other nodes with IP-level edges, otherwise it would have not been detected and it is likely connected to other PoPs, but such PoPs or links were not measured by iPlane or DIMES. For many of the PoPs, there are no other marked PoPs in their vicinity to allow crawling, thus creating "islands" of unmarked PoPs. Last, some PoPs fail the relocation by delay only, mostly because their marked neighbor PoPs do not allow multilateration, e.g., if their (three) neighbor PoPs are co-located.

2) Algorithm's Parameters: As several thresholds are involved in the algorithm, it is important to check their effect on its performance. Our goal is to maintain the accuracy of relocation while minimizing the number of relocation failures. The first thresholds to be tested are those controlling the selection of high level of confidence PoPs: P_{tot_th} and P_{IP_th}. Figure 2 demonstrates the algorithm's sensitivity to these parameters, with the solid line showing the number of PoPs marked during the "Additional Anchor PoPs Marking" step as a function of P_{IP_th} and the dashed line showing the effect of P_{tot_th}. For both thresholds the number of anchored PoPs linearly grows as the threshold decreases. However, even when selecting the most conservative values, meaning setting $P_{DATA\_TH} = 100\%$ and $P_{tot\_th} = 100\%$, which provide both a highest level of accuracy, enough PoPs are marked to use as anchors in the crawling process. The threshold D_{co_th} is evaluated in the same manner, testing the stability of the "Co-Locate PoPs" stage. Table II shows the percentage of Co-Located PoPs as a function of D_{co_th}. We set $D_{co\_th} = 0$, the most conservative value possible. Testing the algorithm's sensitivity to D_{co_th}, varying its value from zero to 5mS, we find up to 4% variance in the number of co-located PoPs both in 2010 and 2012. In 2010 increasing the threshold sometimes reduced the number of co-located PoPs. This is caused by the crawling nature of the algorithm, as a PoP that was marked as co-located in a late iteration is now marked in an earlier iteration, for example in the Location by Delay and Geolocation stage, as one of its neighbor PoPs was marked for co-location using the higher D_{co_th}.

It is possible to find errors in the location of PoPs with a high level of confidence. Such can occur if all databases share the same error. The algorithm searches for such errors during the **Co-Locate PoPs** stage. Having a PoP co-located with multiple marked PoPs with a different location indicates that the marked PoP location is incorrect and should be unmarked. Several such events were flagged and the affected PoPs were unmarked and relocated, being treated as PoPs with a low level of confidence. These cases happened within ISPs such as Cogent (AS174), which were shown [25] to have a large number of false locations in the geolocation databases. The algorithm is thus tested using only anchors with a definite known location. We find that the PoPs marked as anchors during the **Additional Anchor PoPs Marking** stage fall within one of two categories: the algorithm either keeps them in their original place (i.e., by co-location), or fails to relocate them, as they have no marked neighbors. In 2010, these PoPs were 37.5% of all anchor PoPs, while in 2012 they were 70.3%. Consequently, the overall number of PoPs that fail due to lack of neighbors or marked neighbors rose in 2010 by 240% when PoPs with a high level of confidence were not used, while in 2012, the usage of the **Additional Anchor PoPs Marking** contributed only 1% of additional relocation rate.

As many previous works have shown [17], [15] delay measurements for multilateration purposes tend to be inaccurate due to additive latency. The use of PoP level links allows to aggregate multiple edges into a single PoP link and reduce the measurement inaccuracy, as shown in Figure 3. The Figure shows the minimal PoP level link delay in the 2010 dataset compared to the median delay measured on each of its edges (The PoP level minimal link delay is the minimum of all median edge delays). The spread of edge level delays per PoP shows the importance of aggregating multiple edges into a single link. Thus, if a PoP level link is comprised of a single edge, and this edge was measured only a few times, its latency

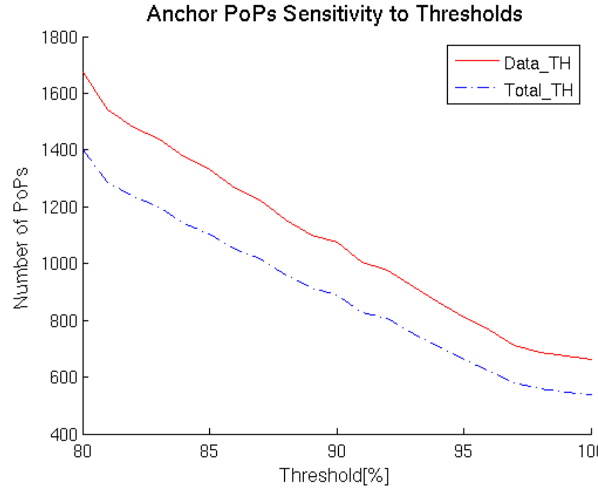


Fig. 2: Number of High Level Confidence PoPs vs. Thresholds Values

Year	D_{co_th} [mS]					
	0	1	2	3	4	5
2010	28.3%	28.7%	26.5%	26.6%	23.8%	23.8%
2012	20.1%	22.9%	23.7%	23.8%	23.9%	24.3%

TABLE II: Number of Co-Located PoPs vs. Threshold Value

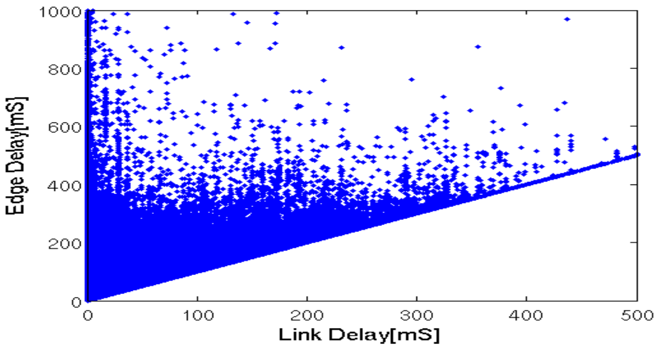


Fig. 3: Link Delay vs. Edge Delay

is likely not to be sufficient from multilateration purposes. We note that 37%-39% of the PoP links contain a single edge, and over 94% of those are measured less than five times. Such links are not used for geolocation by multilateration, as they might introduce large errors.

We rigorously checked the thresholds used for multilateration, but can report only partly due to lack of space. RA_{opt} is selected based on delay to distance ratio on links between anchor PoPs. The ratio of the 2012 dataset is more stable and less sensitive to a change of the thresholds than in 2010 with the average ratio always being in the region of 1.3ms/100km-1.5ms/100km. $RA = 1ms/100km$ is a commonly used value but was shown to be an under-estimate [11], we thus set $RA_{min} = 0.95ms/100km$ and $RA_{max} = 2ms/100km$. As fiber infrastructure depends on terrain conditions and obstacles bypass it is expected that the routed fiber length will be closer to $\sqrt{2}$ air distance, which complies with the ratio measured on 2012. We thus select $RA_{opt} = 1.44ms/100km$.

The values of RA_{min} and RA_{max} do not have a considerable effect on the relocation of PoPs: for $RA_{min} = 0$ and $RA_{max} = \infty$ the PoPs relocated by delay only have an average delay to distance ratio of $1.002 \times RA_{opt}$ and only 8.5% of these PoPs have a ratio outside the default (RA_{min}, RA_{max}) range. The maximal measured ratio is 204 and the minimal is 2.1.

Since the algorithm is oblivious to the multilateration algorithm used, and less than 8% of the PoPs are relocated by multilateration, we refrain from further analysis of this aspect, which was studied by [11], [15], [17].

3) Validation of Location Assignment: When examining the Crawling algorithm location, we need to verify two points: the algorithm must not damage the location of correctly assigned PoPs, and it should correct the location of PoPs that were wrongly assigned. Since the initial location of PoPs is already verified to be very good [6], by keeping the damage close to zero, any improvement in the location of wrongly placed PoPs will result in a very accurate map. The lack of ground truth make geolocation validation difficult, but as we show below, we manage to show that indeed the crawling algorithm performs well.

First, we compare the location assigned to PoPs that we already verified in previous works, and find that relocation assignments are within 200km range. Next, we focus our efforts on ASes where geolocation issues exist, e.g., where the geolocation databases assign all the PoPs to a single location. Validation of ISP's PoPs is done based on the service providers maps. To this end, we use providers maps that were collected by the Internet Topology Zoo project [16] at the same period as our dataset or published by the ISP: Abilene, UUNET (AS701 through AS703), China Telecom, both within China and International, and more. The validation shows that most PoPs are located where expected, with the only exception applying to PoPs placed using multilateration, which are sometimes located with an error of a few hundred kilometers.

Next, we check the correctness of the algorithm using primary anchor PoPs. By unmarking a primary anchor PoP and applying the crawling algorithm to it, one can verify that the location assigned to this PoP is correct and that the algorithm does not relocate PoPs away from their correct location. To this end, we tested 180 primary anchor PoPs. 124 of the PoPs were assigned a location using co-location, 20 PoPs were relocated using *Location Update by Delay and Geolocation Data*, and the rest were assigned a location in the *Location update by delay* stage. Table III shows the breakdown of these PoPs relocation by crawling. Most of the PoPs (82%) retain their original position or are located within 100km from their original location (5%). All the PoPs that are relocated using co-location maintain their original position. A total of 94% of the PoPs are located within 500km range of their original location.

We examine the PoPs that were relocated with an error larger than 500km, and find that the cause is noise in the dataset. The PoPs that were located by *Location Update by Delay and Geolocation Data* are characterized by lack of location in most of the Geolocation databases. For example, Harvard's GigaPoP (AS10578) does not have location information in three of the geolocation databases at all. In one database (NeuStar) a single location appears for all the IP addresses, matching the original PoP location. In the last database, location information appears only for some of the IP addresses. The location information differs from NeuStar's and also points to two different locations, one of them later selected as the new relocation position. This PoP is also characterized by noisy link delay to neighboring anchor PoPs, manifesting as long link delays (hundreds of milliseconds) to PoPs located within a few hundreds of kilometers from Harvard. We find that also within the AS there are long link delays that reach almost 30mS, even between IP addresses in the same CIDR/26. The combination of disinformation in the geolocation database and noisy delay measurements leads to the algorithm's error. Similar noisy delay measurements also affect the PoPs located in the *Location update by delay* stage.

One way to clean noisy link delay is to increase the threshold of required edges between two PoPs above one for the *Location Update by Delay and Geolocation Data* and *Location update by delay* stages. As the median number of edges between a pair of PoPs is two, with an average of

Crawling Stage	Number of PoPs	Same Place	Within 100km	Within 500km	Beyond 500km
Co-Location	124	100%	0%	0%	0%
Delay and Geo-Data	19	16%	47%	16%	21%
Delay Only	37	54%	0%	25%	21%
Total	180	82%	5%	7%	6%

TABLE III: Known PoPs Relocation Accuracy

19.7, this is a conservative rule. The advantage is an increased accuracy, while the disadvantage is the decrease relocation success rate. For example, increasing the number of edges threshold to two, reduces the number of known PoPs relocated from 180 to 145. Out of the 12 PoPs that were located with an error range larger than 500km, 25% of the PoPs are now correctly relocated (within 100km range) and 25% more can not be crawled. Increasing the number of required edges between a pair PoPs to at least three, corrects the location of two more PoPs, thus eventually only 5 PoPs (2.8%) are located outside 500km error range.

The multilateration method used for the geolocation assessment was constraint based. To evaluate possible improvement by using other multilateration methods, we use Spotter [17]. Due to Spotter's resources limitations, we were able to evaluate only the location of the 12 PoPs that were located outside the 500km range, which include 980 IP addresses. Spotter provides a location to 88% of the IP addresses, covering all PoPs. Using Spotter, the location of 58.3% of the PoPs is set within 100km of their true location, 25% more within 500km range, and only 2 PoPs are set outside the 500km region. One of these PoPs is located even further than was estimated by the constraint based approach. We note that Spotter does have some accuracy leverage in the geolocation of the 12 PoPs under study, as most of them belong to universities with, or very close to, PlanetLab [2] nodes. This increases the measurement's accuracy compared to a target located far from PlanetLab nodes.

An advantage of Geolocation using PoPs rather than other methods is shown when considering Spotter's results on the IP level: 41.7% of the IP addresses are located beyond the 100km range of error, and 3.5% beyond the 500km range. Most of the IP addresses outside the 500km error range are located far from adjacent IP addresses, though the PoP to which they belong is located correctly. On one extreme case, of Hong-Kong University, 86 out of 91 IP addresses are located correctly in Hong Kong, three more are within 500km range, one in the Philippine Sea and one far off in Zimbabwe.

An example of an AS where the PoP Crawling algorithm has a significant effect is Telefonica. In Telefonica (AS12956) at 2010, 26 pops are detected (using DIMES data only), and all are originally assigned to Madrid, Spain. After running the PoP Crawling algorithm, the PoPs are assigned to 16 different locations, including Santiago, Chile, Amsterdam, the Netherlands, and more. At 2012, 21 PoPs are detected. The Telefonica PoP map misses some business users' PoPs and some of the Latin America PoPs, but is otherwise accurate.

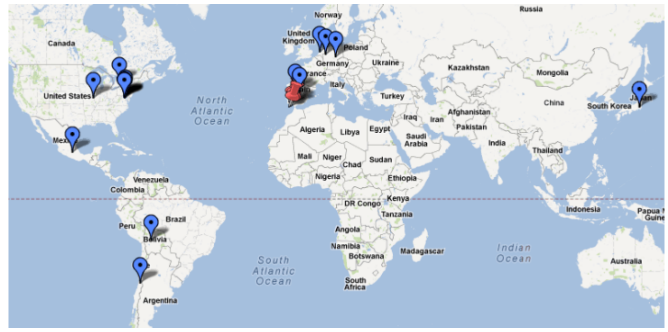


Fig. 4: Telefonica PoPs Location Map

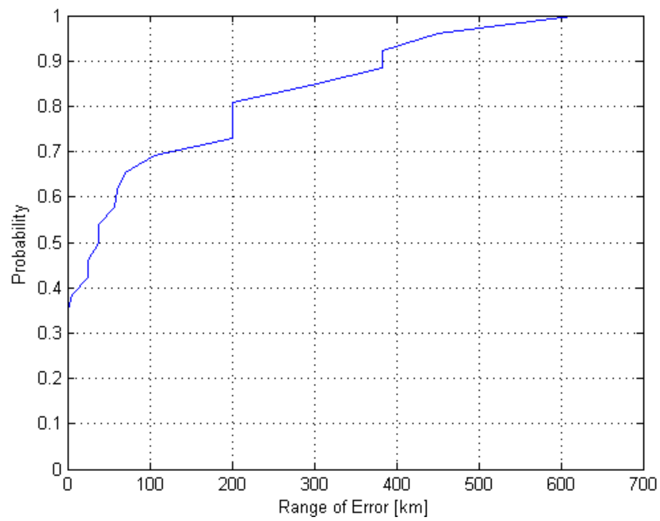


Fig. 5: A CDF of a Large ISP Relocation Range of Error

Figure 4 shows the PoP level map that was validated, with the red pin indicating the PoPs location before running the algorithm, and the blue icons showing the location of the relocated PoPs.

We also corroborated the data with one large ISP and the error range CDF is depicted in Figure 5. 65% of the PoPs were located by the algorithm within 100km from their true location, and 85% within 300km range. Less than 4% of the PoPs are not located within 500km from their true location and in only one case there is a country-level error, where a PoP is located close to the Chilean border. The ISP indicated that before the crawling algorithm was executed, only 23% of the PoPs were placed within 500km range of their true location.

4) Comparison to Geolocation Databases: The effect of the crawling algorithm on a PoP's location is demonstrated in Figure 6. The figure presents a heatmap of the median distance between all the geolocation services used with each dataset and the PoP geolocation algorithms, both naive and crawling geolocation, excluding IP addresses that were not marked during the crawling process. Small distances are marked green, whereas large distances are colored in red. As we have shown in our previous work [25], the databases IPligence, IP2Location and MaxMind have high correlation. Due to the majority vote of the naive algorithm, its median distance from these three databases is very small, less than 45-55km. In

	IPligence	MaxMind	HostIP.info	IP2Location	GeoBytes	GeoPoP Naïve	GeoPoP Crawling
Ipligence	0	7	369	15	353	45	429
MaxMind	7	0	104	7	162	50	467
HostIP.info	369	104	0	254	345	320	804
IP2Location	15	7	254	0	0	49	425
GeoBytes	353	162	345	0	0	283	855
GeoPoP Naïve	45	50	320	49	283	0	444
GeoPoP Crawling	429	467	804	425	855	444	0

(a) 2010 Dataset

	IPligence	MaxMind	HostIP.info	Neustar	DB-IP	GeoPoP Naïve	GeoPoP Crawling
Ipligence	0	13	479	53	224	48	337
MaxMind	13	0	125	21	132	45	341
HostIP.info	479	125	0	331	570	256	926
Neustar	53	21	331	0	329	54	188
DB-IP	224	132	570	329	0	140	619
GeoPoP Naïve	48	45	256	54	140	0	80
GeoPoP Crawling	337	341	926	188	619	80	0

(b) 2012 Dataset

Fig. 6: Heatmap of Median Distance Between Geolocation Services

the 2012 dataset we observe that Neustar's database is also correlated with MaxMind and IPligence, and consequently close to the location by the naive algorithm, as well.

The crawling algorithm leads to a median displacement of PoPs by over 400km (compared to the naive algorithm) in 2010 yet only 80km in 2012. This points to a possible improvement in the geolocation databases. The crawling also results now with locations closer to those indicated by IPligence and Maxmind (compared to 2010). From all the databases, the crawling results are closest to Neustar, which is priced highest from all the geolocation services in use. The relatively small median distance (region range), is a possible indicator of the database's accuracy.

5) IP Geolocation: The contribution of IP-level geolocation using PoPs manifests mainly in the first four stages of the IP geolocation algorithm, thus we evaluate the coverage of IP addresses by these stages as presented in Table IV. Of all the IP addresses measured by DIMES and iPlane, 50% to 60% can be co-located within 2-hops from a PoP with an overall delay of less than 2mS (up to 1mS per edge). As not all the IP addresses are targeted for measurement, it is important to consider also the number of routing blocks (address prefixes) covered by this range. 90525 routing blocks (out of 219750 routing blocks, as indicated at the time by Routeviews [21]) were covered in 2010 and 87785 routing blocks (out of 260954) were covered in 2012.

IP Geolocation Algorithm Stage	2010	2012
PoP Located IP Addresses	87.3K	98.65K
CIDR /24 Blocks of PoP's IP Addresses	38.56K	45.96K
One-Hop IP Addresses	729K	441K
Two-Hop IP Addresses	1004K	672K
Two-Hop Routing Blocks	90525	87785
IP Addresses Measured	1.69M	1.32M

TABLE IV: IP Geolocation By Algorithm's Stage

We next use a set of IP addresses with a known location. They are taken from a 2010 ground truth database provided by CAIDA, described in [14], includes private data from one tier-1 and one tier-2 ISPs. In addition it contains public data from five research networks. The geographic location is provided based on host names, with their encoding provided by the ISP and verified. The database covers 25K addresses, but only 2241 are covered by our 2010 dataset (no aliasing used). 2201 of these addresses were wrongly assigned by the original PoP algorithm. 1656 IP addresses were not marked by the crawling algorithm, and out of the remaining 545 relocated IPs 418 were correctly relocated within 100km, and additional 18 within 500km (The ground truth is highly biased to 2 ISPs and is thus not representative [25])

VI. Discussion and Conclusion

Limitations of geolocation approaches are discussed in detail in previous works [11], [17]. Many of these limitations are mitigated by our PoP-level approach, mainly due to the aggregation since it cleans sporadic noise. We aggregate IPs to PoPs before localization, and we aggregate IP links to PoP edges to clean delay errors for multilateration. The input from multiple Geolocation databases helps in cleaning noise, and so does the comparison between the location of collocated anchor PoPs.

A single-edge delay estimation can be noisy due to differences in reaction time of routers and in the return path. We reduce the effect of the latter by aggregating measurements from multiple vantage points. We provide an analysis of PoP level link delays in [26]. However, subtraction of consecutive delay measurement free us from errors caused by far from straight line routing in previous hops.

Estimating the accuracy of a geolocation algorithm is always a hard task, as one needs information from service providers in order to corroborate the geolocation results. For example, we tried to corroborate our results with multiple University ASes where anomalous results were observed, such as Harvard (AS10578) and Columbia (AS14), but failed. A limitation of corroborating geolocation results with service providers is that geolocation databases tend to be biased on ISP level, meaning for some ISP they are more accurate than for others, thus validating with a few ISPs might not provide a good overall picture. The ISPs used in our validation were selected due to inaccuracies detected in the geolocation databases, namely having many of the ISP's IP addresses located in a single place while in fact they are spread around the world.

Lack of ground truth is also a problem during the validation even if the ISP cooperates. As Triukose *et al.*[29] demonstrated in their work, when cellular networks are involved, a user may seem to be located in one country, while in fact he is roaming in a different country. Such limitations can not be addressed by our algorithm.

We validated the algorithm performance using ground truth data from various sources and show that it improves the geolocation, and in some cases significantly (e.g. in the Telefonica case). We are actively looking for additional validation data. We also plan to further study the IP level geolocation by PoP-IP multilateration.

VII. Acknowledgements

We would like to thank Sándor Laki and Tamás Sebök for their assistance in running the Spotter experiments.

References

[1] Quova. http://www.quova.com, 2010.

[2] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, 2003.

[3] Digital Envoy. NetAcuity Edge. $http://www.digital-element.com/our\_technology/edge.html$, 2010.

[4] DIMES. Distributed Internet Measurements and Simulations. http://www.netdimes.org/.

[5] B. Eriksson, P. Barford, J. Sommers, and R. Nowak. A learning-based approach for IP geolocation. In *Passive and Active Measurement*, pages 171–180, 2010.

[6] D. Feldman, Y. Shavitt, and N. Zilberman. A structural approach for PoP geolocation. *Computer Networks*, 56(3):1029–1040, 23rd Feb. 2012.

[7] Geobytes. GeoNetMap. $http://www.geobytes.com/$, 2010.

[8] B. R. Greene and P. Smith. *Cisco ISP Essentials*. Cisco Press, 2002.

[9] B. Gueye, S. Uhlig, and S. Fdida. Investigating the imprecision of IP block-based geolocation. In *PAM'07: Proceedings of the 8th international conference on Passive and active network measurement*, pages 237–240, 2007.

[10] B. Gueye, S. Uhlig, A. Ziviani, and S. Fdida. Leveraging buffering delay estimation for geolocation of internet hosts. In *Proceedings of the 5th international IFIP-TC6 conference on Networking Technologies, Services, and Protocols (NETWORKING'06)*, pages 319–330, 2006.

[11] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of Internet hosts. *IEEE/ACM Trans. Netw.*, 14(6), 2006.

[12] Hexsoft Development. IP2Location. http://www.ip2location.com, 2010.

[13] hostip.info. hostip.info. http://www.hostip.info, 2010,2012.

[14] B. Huffaker, A. Dhamdhere, M. Fomenkov, and kc claffy. Toward topology dualism: Improving the accuracy of as annotations for routers. In *PAM*, pages 101–110, 2010.

[15] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP geolocation using delay and topology measurements. In *IMC'06*, 2006.

[16] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9):1765–1775, October 2011.

[17] S. Laki, P. Mátray, P. Hága, T. Sebök, I. Csabai, and G. Vattay. Spotter: A model based active geolocation service. In *IEEE INFOCOM 2011*, Shanghai, China, 2011.

[18] H. V. Madhyastha, T. Anderson, A. Krishnamurthy, N. Spring, and A. Venkataramani. A structural approach to latency prediction. In *IMC'06*.

[19] MaxMind LLC. GeoIP. http://www.maxmind.com, 2010,2012.

[20] J. A. Muir and P. C. V. Oorschot. Internet geolocation: Evasion and counterevasion. *ACM Comput. Surv.*, 42(1):1–23, 2009.

[21] U. of Oregon Advanced Network Technology Center. Route views project.

[22] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *SIGCOMM '01*, pages 173–185, San Diego, CA, USA, 2001.

[23] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP geolocation databases: unreliable? *ACM SIGCOMM Computer Communication Review*, 41(2), 2011.

[24] A. Sardella. Building next-gen points of presence, cost-effective PoP consolidation with juniper routers. White paper, Juniper Networks, June 2006.

[25] Y. Shavitt and N. Zilberman. A geolocation databases study. *IEEE Journal on Selected Areas in Communications*, 29(9), Dec. 2011.

[26] Y. Shavitt and N. Zilberman. Geographical Internet PoP level maps. In *TMA Workshop*, 2012.

[27] S. S. Siwpersad, B. Gueye, and S. Uhlig. Assessing the geographic resolution of exhaustive tabulation for geolocating Internet hosts. In *Passive and Active Measurement*, volume 4979, pages 11–20, 2008.

[28] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, pages 133–145, 2002.

[29] S. Triukose, S. Ardon, A. Mahanti, and A. Seth. Geolocating ip addresses in cellular data networks. In *Proceedings of the 13th international conference on Passive and Active Measurement (PAM'12)*, pages 158–167, Vienna, Austria, 2012.

[30] B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *NSDI*, 2007.

[31] K. Yoshida, Y. Kikuchi, M. Yamamoto, Y. Fujii, K. Nagami, I. Nakagawa, and H. Esaki. Inferring PoP-level ISP topology through end-to-end delay measurement. In *PAM*, volume 5448, pages 35–44, 2009.