Toward Better Geolocation: Improving Internet Distance Estimates Using Route Traces

Chandrika Jayant and Ethan Katz-Bassett

December 10, 2004

Abstract

Many applications would benefit from the ability to determine the geographical location of relevant Internet hosts. However, IP addresses do not relate directly to geolocations, and location information is not available for most hosts, limiting what applications are currently able to do.

Researchers have proposed multiple location estimation schemes. These approaches have limited accuracy, and most give estimates only over a discrete location space. The most promising and accurate continuous-space technique is Constraint-Based Geolocation (CBG). Central to the quality of CBG's geolocation is the ability to accurately convert Internet delay to geographic distance.

We investigate these conversions, both analyzing the procedure and investigating the accuracy over measurements between PlanetLab hosts. We present key limitations of the procedure. We then introduce modified approaches that address some of these limitations. In particular, we investigate how knowledge of the route taken to a target might be used to improve distance estimates. We present Path-Based Estimation and Router-Based Estimation, techniques that use this information. We analyze the performance of these CBG-compatible techniques using PlanetLab data. Finally, we present ideas on future improvements to CBG.

1 Introduction

Geolocation refers to the process of inferring the physical location of an Internet host. There is no direct correspondence between geographic location and host IP address, so the best people can do is estimate the location of a host.

The ability to determine a host's location can enable a number of applications. Websites and services could automatically provide location-specific information, such as local weather and news. An online Emergency Broadcast System counterpart could alert users accessing certain high-traffic sites of impending regional emergencies. Applications could automatically localize or restrict their content. Internet advertisers could target ads towards users in specific locations. In fact, some advertisers already attempt to target ads geographically, to mixed results. For instance, advertisements on webmail sites use a mail account's address; one of the authors of this paper was recently confronted with an advertisement for low mortgage rates in Massachusetts, his state of residence when he opened the account, but no longer.

Different levels of granularity in the accuracy of estimates could benefit very different applications. The current best geolocation technique, Constraint-Based Geolocation (CBG), gives a median error of around 100 km and usually gives a confidence region around the size of a midsize U.S. state [4]. However, even the difference between a state and a county is quite significant. For example, an estimation that is accurate to the county-level would allow appropriate delivery of county-specific information, such as election results, store specials, weather reports, and school closings; this type of information would not be appropriate for a state-wide audience.

Is it possible to improve on geolocation to make it useful for future applications that require higher levels of accuracy? We believe that CBG's ability to geolocate is restricted by its inaccuracy in estimating distances given round-trip time delays. This is a difficult problem because the distance a packet travels depends not just on the distance between source and destination but also on the route the packet takes. In particular, a great-circle path rarely exists between a source and destination. Furthermore, the problem is complicated by the fact that RTT includes not just propogation delay but also factors include queuing delay. We propose that Constraint-Based Geolocation could be improved if it used information about the route to a target when estimating the distance to a target. We give two techniques, Path-Based Estimation and Router-Based Estimation, that use route information to improve estimation.

The following section is devoted to a description of the Constraint-Based Geolocation algorithm. Section 3 addresses some problems in CBG's distance-to-delay conversion. In Section 4 we present our approach to fix some of these problems. We lay out our experiments and results in Section 5. Section 6 mentions some related work. We discuss ideas for future work in Section 7 before concluding in Section 8.

2 Constraint-Based Geolocation

Constraint-Based Geolocation (CBG) locates Internet hosts using a set of landmarks, hosts with known geographic locations from which it can make measurements [4]. The Global Positioning System (GPS) [2] measures the distance to a target from multiple vantage points, then performs multilateration using these distances to locate the target. Triangulation is the special case of multilateration from exactly 3 vantage points. Given exact distances from its landmarks to the target hosts, CBG could perform exact multilateration, just as GPS does. Since no way exists to determine the exact distance to the target, CBG instead estimates the distance from a landmark to the target using delay measurements from the landmark to the target. It then performs multilateration using these estimates, leading to a region in which the target is suspected to lie and to an estimated geographic location of the host.

To achieve these distance estimates, each landmark first measures the delay to all other landmarks. Since the landmarks have known locations, each landmark finds a scatter plot relating the geographic distance to each other landmark to the measured delay to that landmark. Figure 1 shows sample scatter plots from our experiments. The baseline represents the speed bits travel in fiber optic cable, 100 km per 1 ms RTT. If a landmark had a direct fiber connection along the great-circle path to each other landmark, we would expect the data points to fall on this line. Since this type of connection is the ideal, no points should occur below it; the distance from source to destination can be at most 100 km per 1 ms RTT between them.

A landmark then fits a "bestline" to its scatter plot. It will use this line to estimate distance to a target from the measured delay. The bestline calculated has the following properties:

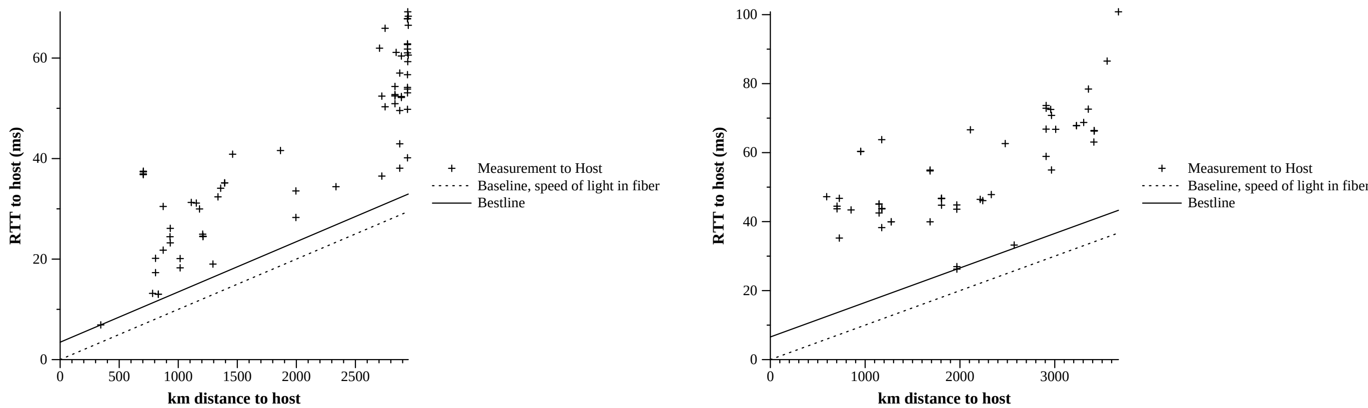
1. The line has nonnegative y-intercept, because it always takes RTT>0 to travel any distance.

2. The line never falls below the baseline, because the baseline is an ideal we cannot beat.

3. The line is fit not to be above any of the points measured from the landmarks. It does not underestimate the distance to any landmark. This fit is chosen because CBG locates targets in the intersection of circles with radius equal to the estimated distance, so the radius should be at least the actual distance. CBG hopes that a fit that does not underestimate the distance to landmarks also will not underestimate the distance to later targets.

4. The line obeys the above properties while fitting as closely to the points as possible.

The calculation of the bestline with these properties is formulated as a linear program. We use D_{ij} to represent the actual geographic distance between hosts i and j and d_{ij} to represent an estimation of that distance. Each landmark L_i finds the geographic distance D_{ij} and RTT delay r_{ij} to each landmark L_j with $i \neq j$. The solution to the program is the slope m_i and the intercept b_i defining the bestline $y = m_i x + b_i$ for L_i. Property 1 gives a constraint $b_i \geq 0$, and Property 2 gives $m_i \geq m$, where $m = 1ms/100km$ the inverse of the speed of light in fiber. Property 3 above gives a set of



Bestline graph for planetlab1.cs.uiuc.edu

Bestline graph for planetlab1.arizona-gigapop.net

Figure 1: Bestline Graphs of Two Landmarks

constraints defining the region of feasible solutions:

$$y - \frac{r_{ij} - b_i}{D_{ij}} x - b_i \geq 0, \forall i \neq j,$$

with the slope $m_i = (r_{ij} - b_i)/D_{ij}$. The objective function to minimize comes from Property 4 above:

$$\min \left(\sum_{i \neq j} y - \frac{r_{ij} - b_i}{D_{ij}} x - b_i \right).$$

The solution m_i and b_i that minimize this function determine the bestline for landmark L_i.

Once all the landmarks have calculated their bestlines, they can geolocate a target host T. Each landmark L_i measures the RTT delay r_{iT} to the target and estimates the geographic distance d_{iT} using its bestline:

$$d_{iT} = \frac{r_{iT} - b_i}{m_i}.$$

These estimated distances d_{iT} are the distances used to multilaterate the location of T. With Property 3 of the bestline, the landmarks attempt to have d_{iT} not be an underestimate. So L_i wants $d_{iT} \geq D_{iT}$ and estimates that T is contained within the circle C_{iT} of radius d_{iT} centered on L_i. Combining the estimates of the N landmarks, T is estimated to be within the region R_T that is the intersection of all the circles (see Figure 2):

$$R_T = \bigcap_{i}^{N} C_{iT}.$$

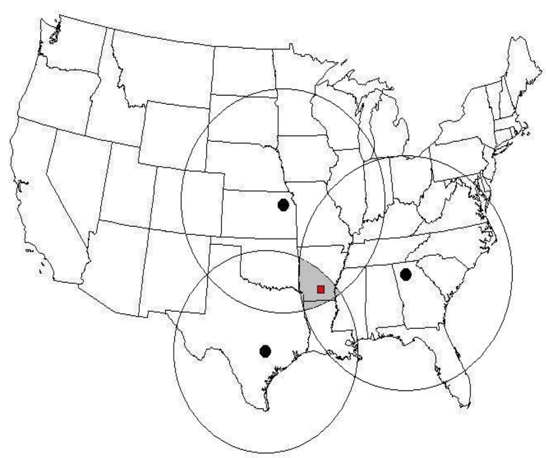


Figure 2: Intersection of CBG Estimate Circles

For the purposes of this paper, we do not concern ourselves with the details of calculating R_T. Instead, we make two arguments about the nature of R_T that do not appear in [4]. First, we note that if, for some L_i, $d_{iT} < D_{iT}$ (meaning, L_i underestimates the distance) then $T \notin C_{iT}$ and therefore $T \notin R_T$. So, whenever a landmark underestimates the distance to T, CBG fails to properly locate T within a region. Second, we note that, for CBG to correctly locate T within a region R_T of radius less than k [1], $D_{iT} - d_{iT} < k$ for some i. Otherwise, no circle C_{iT} comes within k of T, meaning the circle of radius k centered on T is contained in C_{iT} for all i and hence contained in $\bigcap_i^N C_{iT} = R_T$. So, for CBG to locate T within a tight region, at least one landmark must calculate a tight distance estimation to T. So, good distance estimations are necessary for CBG to perform well.

CBG has a number of appealing attributes. First, it assigns locations from a continuous space. Since actual geographic space is continous, this location space makes sense. However, other proposed systems [6, 7, 12] use discrete spaces, in fact assigning locations from a finite set; we discuss such systems in Section 6. The use of a discrete space artificially collocates hosts that actually reside in different locations. Second, CBG calculates fairly accurate location estimates. In experiments reported in [4], CBG bests a discrete space approach, achieving median location error distances below 100 km for a U.S. dataset, versus median error distances of around 150 km for the discrete approach. Third, CBG includes mechanisms that can potentially improve and/or update its estimates. Additional landmarks can be added, giving more distance estimates and potentially more accurate or more tightly constrained location estimates. The original landmarks can retrain on the original set and/or train on additional landmarks in order to achieve new bestlines that better reflect the current state of the network. In Section 3, we consider problems with CBG and, especially, with CBG's landmark-based bestline estimates.

3 Problems with Landmark-based Bestline Estimation

As explained above in Section 2, each landmark in [4] calculates a single bestline and uses it to estimate the distance to targets. In this section, we describe problems with this approach. Based on how the bestline is calculated and used, we argue that these problems exist, and in Section 5 we experimentally verify that the problems exist.

[1] Meaning, R_T contains T but does not contain any points distance k from T.

Before we discuss specific problems, we note that the bestline of a landmark L_i is generally determined completely by measurements to at most two other landmarks, say L_j and L_k. By this observation, we mean that, usually:

1. For all $m, j \neq m \neq k$, L_i's bestline would not change if it did not have measurements to L_m.
2. If L_i did not have measurements to either L_j or L_k, its bestline would change.

These observations are based on the fact that the bestline is computed to be less than or equal to all data points, but to be as close to the data points as possible. So the bestline usually ends up going through the lowest two points. Removing a point higher than these two does not affect the bestline; it still cannot be above the two points. However, removing one of the two low points usually causes the bestline to change in order to be as close to the data points as possible.

The statements are not always true because of two special cases. First, L_i could have three or more colinear measurements that all lie on the bestline. Second, L_i's bestline could be determined by only one point, say the measurements to L_j. This case occurs when, for all $k \neq j$, the line through both L_j's and L_k's measurements either has slope less than $1ms/100km$, has negative intercept, or does not fall below all the data points. Even in these two special cases, the bestline for any landmark is mostly likely determed completely by only a few measurements.

We now describe four problems with this approach in which each landmark calculates all distance estimates using a single bestline. We refer to this approach as Landmark-Based Estimation (LBE). Most of these problems occur because the bestline is calculated to be below all data points or because, as we noted, the bestline usually is based on only a few measurements.

3.1 Landmark-based bestline estimates are often not tight.

The bestline is fit to be below all known data points. In particular, it is fit even fit to be below any low outlier points the landmark might measure; consider, for instance, the bestline for the *planetlab1.arizona-gigapop.net* landmark shown in Figure 1. Clearly, the bestline is fit to two outlier points and has no relation to most of the points. CBG is based on the idea that the measurements from a landmark to the other landmarks are fairly representative of the measurements that that landmark should expect to see to targets. Therefore, we expect the distribution of data points from targets to be somewhat similar to the distribution of data points from landmarks. We would expect, then, that the bestline for *planetlab1.arizona-gigapop.net* would lie below actual measurements for most targets. Then, the landmark *planetlab1.arizona-gigapop.net* would significantly overestimate the distance to most targets.

3.2 In general, landmarks cannot even accurately estimate the distance to other landmarks.

As we argued above, in most cases the bestline for a landmark L_i depends only on measurements to a few other landmarks. The bestline does not depend on measurements to any other landmark L_m that L_i trained on, so we would not expect L_i to be more accurate in estimating the distance to L_m than it would be in estimating the distance to an arbitrary target. Now, consider a target T that is not a landmark itself but is collocated with L_m; say, T and L_m are both hosts in the room in the Allen Center. Clearly, the distances from L_i to L_m and to T are essentially the same. We expect the delay to be very similar as well. By calculating the delay and distance to known landmarks such as L_m, L_i is meant to train its delay-to-distance conversion. We would expect a trained estimator to perform better on values similar to those it trained on. However, because the bestline for L_i does not depend on measurements to L_m, we do not expect L_i to be better at estimating the distance to this collocated T than it would be at estimating the distance to an arbitrary target. It is a queer property of bestline training that L_i does not learn anything from most of the landmarks it trains on.

3.3 Training on additional landmarks does not increase the accuracy of distance estimations.

As defined, the bestline for a landmark L_i is calculated to be below all measured points. If L_i makes distance and delay measurements to an additional landmark L_k not in its original training set, there are two possibilities. First, the measurements for L_k could fall on or above the bestline. In this case, the bestline will not change, and the landmark's estimations to targets will remain the same.

Second, the measurements to L_k could fall below the bestline. In this case, the bestline will be recalculated to be below this new point. Assuming again that the measurements from L_i to other landmarks are fairly representative of the measurements from L_i to targets, we expect the measurements for most targets to fall above the bestline (indeed, the bestline is supposed to fall below them in order to produce overestimations of distances). So, when the bestline is recalculated, it will tend to move down, moving farther away from the measurements to most targets. This means that estimations calculated with this new bestline will tend to be less accurate than estimations made with the previous bestline, before measurements were taken to L_k.

3.4 Bestline estimates underestimate some distances.

The bestline for a landmark L_i can only equal the baseline if, for some landmark $L_j, i \neq j$, $r_{ij} * 100 = D_{ij}$ (the delay times the speed of light equals the distance). This would imply that there is a great circle fiber connection between L_i and L_j and that L_j requires no processing time to reply to L_i's ping. So for any actual set of landmarks, no bestline can equal the baseline. So, for any L_i, there is at least a conceptual chance that, for some target T, the measurements r_{iT}, D_{iT} fall below the bestline and above the baseline. So L_i would underestimate the distance to T. If, in practice, some landmarks actually do underestimate the distance to some targets, then CBG ought to be able to deal with underestimations.

4 Our Approach

This section outlines our approach to improve upon CBG's distance to delay conversions.

4.1 Motivation

As discussed in the Introduction, converting delay to distance is a difficult task. First, the delay of a packet depends on the route it takes, not just on the distance from source to destination. Second, delay includes not just propagation time but also factors such as processing and queuing delay.

We believe that the general approach used by CGB - of converting delay to distance and then performing multilateration - is a good one. However, we believe that the delay to distance conversion is too simple to provide accurate results. We propose possible improvements and analyze the success of some of these improvements through a set of experiments. The left graph in Figure 1 shows the bestline from *planetlab1.cs.uiuc.edu* to its targets. There is a large cluster of points around the 2500 km mark, representing hosts that are all approximately the same distance from *planetlab1.cs.uiuc.edu*. However, the delay ranges from 50 to 70 ms, and a single bestline cannot accurately capture this.

We hypothesize that targets with similar routes have similar delay to distance conversions. For example, the routes from a UW host to Harvard and to MIT (both in Massachussetts) will be much more similar than routes to Harvard and, say, to Stanford (in California). By using information about the route to a target, we believe landmarks can achieve more accurate distance estimates. The two new techniques we introduce are Path-Based Estimation and Router-Based Estimation. Both techniques fit into the CBG multilateration framework. In both of these techniques, we still use bestline fitting, but future work should include experimenting with other line fits.

4.2 Path-Based Estimation

In Path-Based Estimation (PBE), a landmark learns routes to its training set. It *traceroutes* the targets with TTL up to x. The landmark then finds the longest prefix of this partial route that is shared

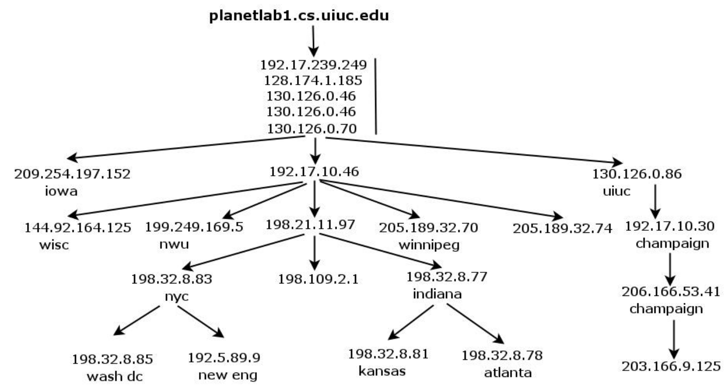


Figure 3: Routes from planetlab1.cs.uiuc.edu

with a subset of its training hosts. The landmark then calculates a bestline, using only this subset. Below, we refer to the first x hops of a route as the "x route-prefix" of the route. In Figure 3, we show some routes from the University of Illinois- Urbana Champaign (for 9 hops). By looking at this diagram, it seems clear that many routes are directional. Routers branch off to other locations in different directions and seem to span a small part of the country. For example, one path from New York City goes towards Washington D.C., while the other goes the other direction towards New England. Indiana splits off to Kansas and Atlanta. And both Indiana and New York City were in completely different directions from Winnipeg, Northwestern University in Illinois, and Wisconsin- all children paths stemming from Illinois.

4.3 Router-Based Estimation

Now we propose a system in which a landmark need not measure all of the x route-prefix. In Router-Based Estimation (RBE), a landmark learns routes to its training set, as in Path-Based Estimation. It then sends packets to the targets with TTL = x. The landmark finds a subset of training hosts with paths through this router, and then calculates a bestline using only this subset. In practice, we advocate using a sequence of TTL's instead of a single one. Our experimental results below motivate this.

5 Experimental Results

In this section, we describe our experimental results. First we discuss our test environment and datasets. Then we discuss the results of specific experiments.

5.1 Environment and Datasets

In our experiments we used PlanetLab and Scriptroute to gather data. PlanetLab is a collection of machines distributed around the globe, currently consisting of 446 machines, hosting by 209 sites, and spanning over 25 countries [10]. Most PlanetLab machines are hosted by research institutions, with others located in co-location and routing centers. All of the machines are connected to the Internet. Using PlanetLab routers gave us access to geographically dispersed machines from which to conduct our experiments. Using 110 of PlanetLab's hosts, mostly located in the U.S. and some in Canada, we had a large, real world test bed on which to run our tests. 26 of these hosts became our landmarks, after munging (combining very close landmarks into one landmark with more potentially successful source-target measurements).

Scriptroute is a flexible network measurement system built in 2003 at the University of Washington [11]. There are currently 307 operational servers in the system, most of them also part of PlanetLab. Scriptroute allows a user to connect to any server and execute any safe network measurement, providing us with an easy mechanism to execute the measurements across a range of hosts.

All PlanetLab hosts have a known geographic location (latitude and longitude). Scriptroute was used to measure delay using *ping* and gather routes using *traceroute* between pairs in our set of hosts. We ended up with 1411 measurements between 26 landmarks (munged) and 96 targets. Our landmarks were spread all across the United States, spanning the magor regions (west, east, north, south, and central). There were 2 landmarks in Canada near the border. Figure 4 shows a map of our landmarks.

To collect data, we remotely ran versions of *ping* and *traceroute* between the members of the set of hosts we had available. We did this over a few days to get as many measurements as possible (at the time of our experiments, many PlanetLab machines were being upgraded from version2 to version3, causing many machines to temporarily be unavailable). Since the targets don't need to be running Scriptroute, in the future we would like to test on more geographically diverse targets (even though ours were spread out in the U.S. fairly evenly). Another step in gathering data would be to not just concentrate on U.S. hosts but hosts from around the world. However, it seems that concentrating on one large area is sufficient for the beginning experiments of our route information hypothesis. It would be interesting to see if routes had different patterns in different parts of the world, and how that would affect training the landmarks.

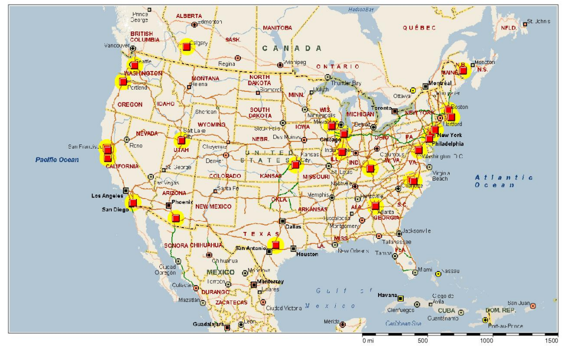


Figure 4: Map of Landmarks

5.2 Performance of Path-based Estimation

In this section, we evaluate the effectiveness of path-based estimation. We also evaluate how the effectiveness changes as we measure longer route-prefixes from landmark to target. We take the entire set S as landmarks and train them on each other. So each landmark has the delay and the entire route to other landmarks. We take $D = H - S$ as our set of targets; all hosts that we have measurements to but not from. We have a total of 1411 distinct pairs $L \in S, T \in D$ such that we have a delay measurement and route trace from L to T. For each technique, we estimate the distance from

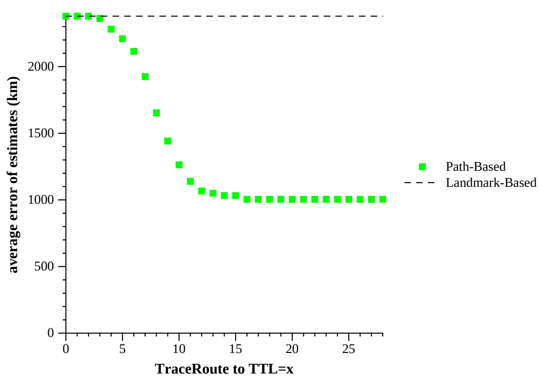


Figure 5: Path-Based Estimation

L to T for each such pair. We take the mean of its estimation error distance over these 1411 estimations as a measure of the effectiveness of a particular technique. We calculate these error distances from the estimated distance and from the known location of each target. We use CBG's standard landmark-based bestline estimation as our point of comparison. We then test the effectiveness of path-based estimation in a sequence of 28 trials (since 28 is the longest route between any pair of hosts in our dataset). In the i'th trial, for each $L \in S, T \in D$, we use our path-based estimation technique at L to estimate the distance from L to T using the length i route-prefix of the route from L to T (or using the full route if it is less than or equal to i hops).

Figure 5 shows our results from these trials. Landmark-Based Estimation has a mean error distance of 2378.4 km. As seen in the graph, path-based estimation always does at least as well. Using 6 hops, for instance, gives a mean error distance of 2114.4 km, and 12 hops gives mean error distance of 1066.9. Furthermore, for all i, the mean error distance using paths of up to $i + 1$ hops is never more than the mean error distance using only i hops of each route. So, at least on this dataset, we never lose accuracy by using more of a route. This fact endorses our intuition that two routes that share a prefix have similar delay→distance conversions. If we know some of the route to a target, we are able to select a subset S_p of hosts from our training set that share that partial route p. It appears that calculating the bestline using only this subset leads to more accurate estimated distances to the target than calculating over all of S.

Note that for $i \leq 3$, path-based estimation performs as poorly as landmark-based estimation. At least on our dataset, most routes from a given landmark were the same for the first 3 hops. So in these cases, $S_p = S$, and we are calculating the same bestline as landmark-based estimation. Paths of length $i + 1$ give significantly lower mean error distances than paths of length i for $6 \leq i \leq 9$. We believe that this occurs for two reasons. First, most routes in our dataset have length at least i for $6 \leq i \leq 9$; if a route is only of length n, considering TTL$\leq n+1$ does not gain us anything, but at these values of i, we learn more about each route with each trial. Second, these hop numbers are when routes tend to diverge from each other based on the region of the country in which the target resides. In our UIUC path example, for instance, on the 7th hop routes that shared the first 6 hops split, with some heading to Wisconsin and some to Winnipeg. On the 8th hop, routes that went together through 7 hops split between New York and Indiana. Paths through New York on hop 8 split on hop 9, heading to New England or Washington, D.C. We hypothesize that these hop numbers are particularly important because they often strongly determine both in which region the target is located and the directness or indirectness of the path to this region. Finally, we note that the mean error distance barely decreases when we consider prefixes of length 13 and does not decrease at all after paths of length 16. There are three causes for this. First, many routes are not longer than 16, so by $i = 16$ we have already matched those routes as much as is possible. Second, for most routes of length greater than 16, our landmarks have likely not trained on any routes that share the first 16 hops. Third, although future investigation is required to verify, we believe that much of the median error at this point is produced by a few particularly bad estimations, so most of our estimations that consider paths of length up to 16 are actually quite accurate and difficult to improve. These bad estimations are caused by very large measured RTTs.

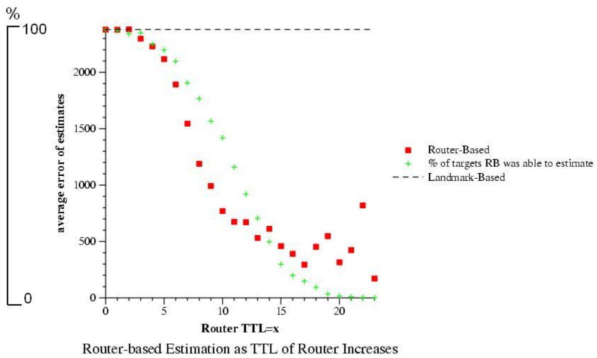


Figure 6: Router-Based Estimation

5.3 Performance of Router-Based Estimation

In this section, we evaluate the effectiveness of Router-Based Estimation. We also evaluate how this effectiveness changes as we base it on routers more hops along in the route from landmark to target. We perform our evaluation using a technique equivalent to that in 5.2. Again, we take the entire set S as landmarks and train them on each other. We take $D = H - S$ as our set of targets, yielding 1411 distinct pairs $L \in S, T \in D$ such that we have a delay measurement and route trace from L to T. For each technique, we estimate the distance from L to T for each such pair. We take the mean of its estimation error distance over these 1411 estimations as a measure of the effectiveness of a particular technique. As above, we use CBG's standard landmark-based bestline estimation as our point of comparison, and we test the effectiveness of RBE in a sequence of 28 trials. In the i'th trial, for each $L \in S, T \in D$, we use RBE at L to estimate the distance from L to T using only the i'th router on the route from L to T. For each trial, we also calculate the percentage of (L, T) pairs, of the 1411 possible, for which RBE can produce an estimated distance.

Figure 6 shows our results from these trials. First, note that Router-Based Estimation never has a higher mean error distance than LBE; at least on average, calculating the bestline using only measurements to landmarks that have a router in common with the target (on the path from the measuring landmark) never worsens estimations. Second, we note that the percentage of (L, T) pairs for which RBE can produce an estimated distance decreases as i increases. The percentage is 0% for $i > 23$. Third, we note that using router i with $i > k$ usually yields better mean error distance than using router k. The values $i > 17$ that yield much worse mean error distance than $i-1$–say, $i = 22$– generally do so because, for such high i, RBE only produces an estimate for a small number of (L, T) pairs. So, one bad estimate greatly affects the mean.

The percentage decreases and eventually goes to 0% for two reasons. First, a route of length $< i$ does not have an i'th router. So, in the i'th trial, RBE cannot produce an estimate for any pair (L, T) where the route from L to T is less than i hops long. Second, a landmark L produces RBE estimates given an i'th hop router R_i by considering landmarks from its training set that have routes through R_i. It is unable to produce an estimate if none of the landmarks have such a route; in effect, L knows nothing about R_i. As i increases, it becomes less likely that L knows any routes through R_i. We can see this by considering that the number of routers i hops from L generally increases exponentially with i.

We conclude that RBE seems to produce better estimates than LBE. In general, RBE using routers only a few hops from L will produce many estimates, but the mean distance error will be high. RBE using routers many hops from L will produce a small number of accurate estimates. These observations motivate the modified version of RBE in which a landmark tries to produce an estimate using a sequence of routers $R_{x_1}, R_{x_2}, \ldots, R_{x_n}$, with $x_i > x_{i+1}$. RBE with R_{x_i} is likely to be more accurate than $R_{x_{i+1}}$ should it be capable of producing an estimate, but is less likely to be able to. If, for a given (L, T) pair, all R_{x_i} fail to produce an answer, the modified RBE uses standard Landmark-Based Estimation. For the rest of our experiments, we use the sequence R_{12}, R_9, R_6. We want to test if RBE can be as effective as PBE only knowing a few of the routers along the path. We choose these routers based on the results described in this section and on our arguments in 5.2 as to the importance of hops 6 through 9. However, the choices are fairly arbitrary, and proper selection of the sequence should

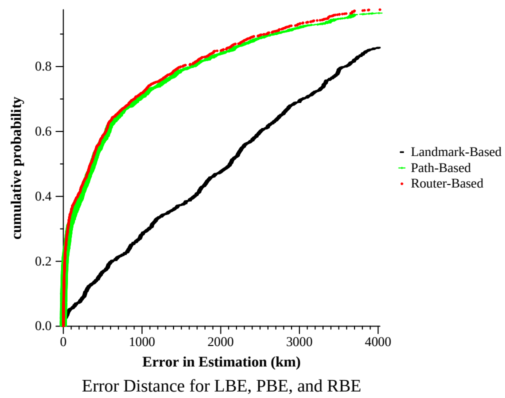


Figure 7: Accuracy of Estimation Techniques

be the topic of further study.

5.4 Comparison of Estimation Techniques

In this section, we experimentally compare LBE, PBE using route prefixes of length 12, and RBE using the router sequence R_{12}, R_9, R_6. A good estimation technique should, with high probability, provide accurate estimates. We again take the entire set S as landmarks and train them on each other, then compare the quality of our techniques on estimating the distances between the 1411 (L, T) pairs in our dataset.

Figure 7 shows, for each technique, the cumulative distribution function (CDF) of the estimation error distance in this experiment. We see that RBE and PBE produce good estimates with much higher probability than LBE. RBE produces estimates that are within 100km of the actual distance with probability 0.35, and PBE produces such estimates with probability 0.32. LBE, meanwhile, only produces them with probability 0.05. With PBE and RBE, we have probability greater than 0.5 to have one after measurements from just 2 landmarks [2]. With LBE, we only have probability greater than 0.5 of a measurement within 100km after 14 measurements [3]. From section 2, we know that, to have any possibility of locating a target within a 100km radius region, CBG must have at least one estimated distance that is accurate to within 100km. We conclude that it is likely to take measurements from many fewer landmarks to achieve tight regions with PBE or RBE than it is with LBE.

With probability 0.5, LBE produces estimates that are off by 2119km or more, validating our argument in Section 3.1. Most of the time, PBE and RBE produce much better estimates than that. In this experiment, with probability 0.5, RBE produces estimates that are within 346km of the actual distance, and PBE produces estimates that are within 376km. Also, the argument in Section 3.4 is reinforced by the observation that LBE underestimates the distance to 100 targets in this experiment.

Perhaps surprisingly, RBE slightly outperforms PBE, even though it uses less information– three routers along a route, instead of the entire route. This issue needs further investigation, but we believe that it is because a landmark L may have multiple paths to a given router R. In this case, PBE will calculate its bestline using only hosts in the training set that match a single path through R, specifically the path shared with the target. RBE, on the other hand, will calculate its bestline using all hosts in the training set with paths through the router; by using more hosts, RBE arrives at more accurate estimates.

5.5 Estimations to Targets Collocated with Landmarks

In this section, we assess the ability of LBE, PBE using route prefixes of length 12, and RBE using the router sequence R_{12}, R_9, R_6 to estimate the distance to targets collocated with landmarks. For our experiment, we again take the entire set S as landmarks and train them on each other. Each landmark then estimates the distance to the other landmarks and to hosts that are collocated with other landmarks. A host is collocated with a landmark if it has the same longitude and latitude. We have 54 targets in our set; 26 landmarks plus 28 hosts collocated with a target.

[2] We see that $1-(1-0.35)^2 > 1-(1-0.32)^2 = 0.5376 > 0.5$

[3] We see that $1-(1-0.05)^{14} = 0.5123 > 0.5 > 1-(1-0.05)^{13}$

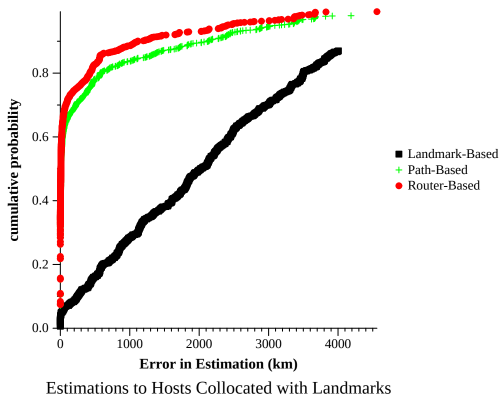


Figure 8: Accuracy to Hosts Collocated with Landmarks

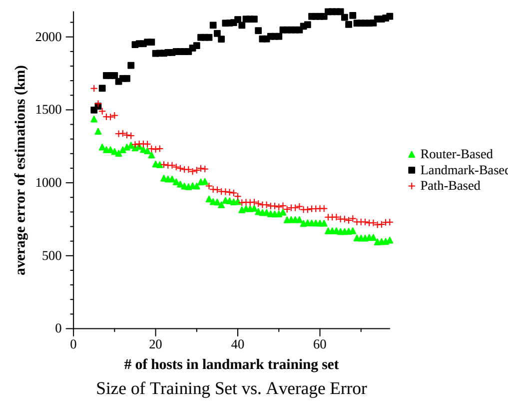


Figure 9: Effect of Increased Training Set Size

Figure 8 shows the results of this experiment. As we predict in Section 3.2, LBE is not accurate at estimating these distances. LBE estimates with error distance less than 100km with probability 0.07, barely better than it does with targets that may or may not be collocated with landmarks. Even though LBE knows the delays and distances to the other landmarks, its bestline does not allow for accurate estimations. Meanwhile, PBE and RBE very frequently make estimations that are within 100km of the actual distance; PBE makes them with probability 0.66, and RBE with probability 0.71. If T is a target collocated with a landmark L_1, then the route from another landmark L_2 to T will be very similar to the route from L_2 to L_1. Knowing the route from L_2 to L_1, PBE and RBE use this knowledge to accurately estimate the distance to T.

5.6 Effect of Increased Training Set Size

In this section, we assess the impact of larger training sizes on the accuracy of the estimations of LBE, PBE using route prefixes of length 12, and RBE using the router sequence R_{12}, R_9, R_6. For our experiment, we again use the entire set S as landmarks. We run 74 trials. In trial i, we randomly select $i+4$ hosts from the set $D = H - S$ of non-landmarks, and we train the landmarks on these $i+4$ hosts. Each landmark then estimates the distance to all other landmarks. We consider the mean error distance.

Figure 9 shows the results of this experiment. As we predict in Section 3.3, increasing the size of the training set tends to decrease the accuracy of LBE. However, the accuracy of PBE and RBE tends to increase as the training set gets larger. As they train on more hosts, PBE and RBE learn more routes and routers, and they learn more about the routes and routers they already know. We see that the marginal utility of adding a host to the training set decreases as the set gets larger; the more routes the landmarks have already seen, the less likely it is that adding one more host with add a significant amount of knowledge. Further study is required to determine the appropriate training set size. However, we can conclude that PBE and RBE tend to get more accurate the more they are trained, and LBE tends to get less accurate.

6 Related Work

In this paper we have concentrated on CBG, pointed out its limits, and suggested improvements. To more clearly understand the bigger problem of geolocation, and why we believe that CBG is a very promising approach, we discuss other related work in the field.

For many applications, geographic distance is not the appropriate metric for making network measurements. For instance, when selecting from a list of mirrors to serve a download, the selected mirror should be "close" in terms of delay and, perhaps, available bandwidth, but need not be "close" in physical distance. Propagation delay increases directly with distance traveled. However, packets usually do not travel in a straight line from source to destination. Furthermore, network delay is variable and includes not just propagation delay but also queuing and processing delay, preventing a direct conversion from delay to distance. Hopcount is another useful network distance measure, but, again, does not necessarily directly correlate with geographic distance. Research has been done to determine appropriate metrics for certain situations and to design systems to support measurement[9, 3].

Researchers have proposed different solutions to the problem of geolocation. Netgeo tries to leverage already available sources of geographic information to determine the location of an IP address, domain or machine name, or AS number[6]. When possible, it combines multiple sources of location information and assesses the reliability of its determination. It also identifies the granularity of its location determination; while some targets might be identifiable as being in a particular city, others might only be linked to a particular country. Netgeo relies mainly on DNS LOC records, whois registration records, and specialized hostname rule files. DNS LOC records can map IP addresses to latitude and longitude. Whois maps domain names, AS numbers, and IP addresses to the location of a registered contact. Hostnames sometimes have naming conventions that include geographic information; for instance, some ISPs include the three letter code of a nearby airport, and Netgeo can extract this information and use it to localize the host. Netgeo's three approachs have various problems. Among others, LOC records are unnecessary and hence rarely provided or infrequently updated. Whois sometimes returns a false location (all subscribers to a large ISP may map to ISP headquarters). Many hostnames do not include geographic information, and some may trigger false-positive hits by inadvertently including, for example, an airport code. IP2GEO encapsulates a suite of three techniques[7]. GeoTrack uses DNS names of the target host or of nearby nodes to guess location (e.g., cs.colorado.edu indicates Boulder, Colorado), similar to Netgeo's hostname mapping. The "nearby nodes" are the closest nodes to the target along the route from various probe machines. GeoPing measures the network delay from a target to a set of host locations with known locations (landmarks), and then it guesses the location of the target to be the location of the landmark with the most similar set of network delays. GeoCluster uses partial IP-to-location mapping data (from sources including webmail account registrations) and BGP prefixes to identify clusters of IP addresses such that each cluster may be located in the same area. Among other limitations, each of the IP2GEO algorithms assigns locations only from a discrete set. [12] investigates the correlation between delay and distance and presents a geolocation technique very similar to GeoPing, again with a discrete set of assignable locations.

Constraint-Based Geolocation (CBG) avoids this limitation, assigning locations from a continuous geographic space[4]. CBG measures delay to a target from a set of landmarks, calculates a delay-to-distance function for each landmark, then performs multilateration to estimate the location of the target. In experiments described in [4], CBG achieved a median error distance for a U.S. dataset of below 100 km.

Since the accuracy of the delay to distance conversion strongly affects the accuracy of location estimates, network delay research is extremely relevant to the problem of geolocation. [8] presents a technique to infer the distribution of the variable delay along an internal link. The Internet Coordinate System (ICS) extracts topological information from delay measurements between beacon hosts [5]. The technique uses these delay measurements to construct a Cartesian coordinate system on which to locate other target hosts. Similarly, [1] builds a decentralized coordinate system (Vivaldi) mapped from the network and uses it to predict RTT. Its main contribution is mapping into a 2D model with height vectors to capture transmission delays on certain access links (single-homed hosts). This approach is interesting because the paper strives to distinguish between different types of delay, and shows how regular 2D coordinate systems do not deal well with distorted latency adjustment. This approach seems very promising and intuitive.

7 Future Work

In this paper, we give techniques to improve CBG's distance estimation. Future work should include numerical analysis of the effectiveness of these techniques. Experiments should include studying the optimal route-prefixes to use in Path-Based Estimation, and the optimal router sequence to use in Router-Based Estimation. PBE and RBE should be fit into a full implementation of CBG, to study the effects of the improved distance estimation on CBG's geolocation. CBG must be improved to better handle underestimations of distance.

We further propose to study the impact of line fits other than bestline. We believe lines that better match all the data (overestimating some distances, underestimating others) may provide more accurate distance estimates.

Distance estimates could further be improved by techniques to separate out components other than propagation delay from the measured RTT. A Vivaldi-like system could allow this. Similarly, probabalistic models of the delay on a given route might be used to estimate the current congestion, allowing PBE and RBE to counteract the delay-amplifying effects of congestion when estimating distance.

8 Conclusions

Geolocation potentially enables a large range of powerful applications. In order for geolocation to be useful, estimates must be much more accurate and consistent.

CBG is an interesting and fairly accurate technique for geolocation, but our analysis and experiments indicate that it could be improved. In particular, CBG's Landmark-Based delay to distance estimation is rife with problems. Using route information significantly improves the accuracy of these estimates. Path-Based Estimation and Router-Based Estimation perform significantly better than CBG's Landmark-Based Estimation scheme, as shown in our experiments. More numerical analysis is needed to confirm and quantify these results, but we are confident in the ability of these techniques.

Because of the poor quality of LBE's distance estimation, CBG requires measurements from many landmarks to accurately geolocate a target. By producing close estimations with high probability, PBE and RBE give the possibility of accurate geolocation using measurements from fewer landmarks. We believe that this makes up for the overhead of route measurements. Furthermore, by tending to produce more accurate estimations as more of the route to a target is measured, PBE and RBE provide a means to tradeoff accuracy for computational overhead. An application that needs extremely accurate geolocation might measure more of the route to a target than one that requires only a regional location. We believe that PBE and RBE provide powerful tools for flexible and accurate geolocation.

References

[1] F. Dabek, R.Cox, F.Kaashoek, and R.Morris. Vivaldi: A decentralized network coordinate system. In *Proc. of the ACM SIGCOMM '2004*, Portland, OR, USA, 2004.

[2] P. Enge and P. Misra. Special issue on global positioning system. In *Proceedings of the IEEE*, volume 87, pages 3–15, jan 1999.

[3] P. Francis, S. Jamin, C. Jin, Y. Jin, V. Paxson, D. Raz, Y.Shavitt, and L. Zhang. Idmaps: A globalinternet host distance estimation service. In *IEEE/ACM Transactions on Networking '2001*, 2001.

[4] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of internet hosts. In *ACM Internet Measurement Conference 2004*, 2004.

[5] H. Lim, J.C. Hou, and C. Choi. Constructing internet coordinate system based on delay measurement. In *ICM '03*, 2003.

[6] D. Moore, R. Periakaruppan, J. Donohoe, and K. Claffy. Where in the world is netgeo.caida.org? In *Proc. of the INET '2000*, Yokohama,Japan, 2000.

[7] V. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for internet hosts. In *Proc. of the ACM SIGCOMM '2001*, San Diego,CA,USA, 2001.

[8] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. In *UMass COMPSCI Technical Report 99-55*, 1999.

[9] S. van Langen, X. Zhou, and P. van Mieghem. On the estimation of internet distances using landmarks. In *Proc. of the International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking-NEW2AN'04*, St.Petersburg,Russia, 2004.

[10] www.planet lab.org. Planetlab website.

[11] www.scriptroute.org. Scriptroute website.

[12] A. Ziviani, S. Fdida, J. Ferreira de Rezende, and O. Carlos Muniz Banderia Duarte. Toward a measurement-based geographic location service. In *Proc. of the Passive and Active Measurement Workshop- PAM '2004*, Antibes Juan-les-Pins,France, 2004.