GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements

M. J. Arif, S. Karunasekera
Department of Computer Science and Software Engineering
University of Melbourne, Australia
marif, shanika@csse.unimelb.edu.au

S. Kulkarni
NICTA Victoria Labs, Australia
santosh.kulkarni@nicta.com.au

Abstract

Knowing the geographical location of an Internet host is of importance to many of today's Internet services. In this paper we focus on geolocating Internet hosts based purely on latency measurements. Existing latency measurement-based geolocation techniques use the observed latencies from multiple landmarks to the target host to determine maximum bound or both the maximum and minimum bounds of the geographical region where the target host is located. Due to the large variance of Internet latency measurements, the region constrained based on such maximum-minimum bounds tends to be relatively large resulting in large estimation errors. We propose a geolocation algorithm, GeoWeight, which improves the geolocation accuracy by further limiting the possible target region by dividing the constrained region to sub-regions of different weights. The weight assigned to a sub-region indicates the probability of the target being in that sub-region; a higher weight indicating a more probable region. By considering latency measurements from multiple landmarks and computing the resultant weights of overlapping regions a better constrained target region can be obtained. This paper presents the GeoWeight algorithm and evaluates its performance using both synthetic and real data by geolocating target hosts in North America. We compare GeoWeight with two popular geolocation techniques, Octant and CBG, by geolocating the same set of targets. The results show that the GeoWeight algorithm outperforms existing techniques.

1 Introduction

Internet host geolocation is an important research problem that is currently addressed by many research groups. Internet location information can be leveraged to improve the user experience and determine business strategy. Some uses of such location-aware systems include geographically targeted advertising on web sites, automatic selection of language to display web site content, web content delivery based on region, credit card fraud detection, and load balancing and resource allocation between Internet hosts.

Our goal is to develop a scalable and reliable Geolocation technique to locate hosts on the Internet. However, what makes this task challenging is there is no one-to-one mapping between IP addresses and geographic locations. The dynamic nature of IP address assignment makes the host geolocation in an IP environment even harder. On the other hand, wireless domain localization is well addressed [20], as the transmission characteristics in the air are relatively regular. Internet host localization is more difficult because the transmission characteristics on the Internet are abruptly influenced by factors such as circuitous route and queueing delay on the routers.

Current measurement-based approaches for geolocation mainly use end-to-end latency measurements from a set of nodes with known location to the node to be geolocated. Nodes with known locations are referred as *landmarks* and the nodes to be geolocated are referred to as *targets*. Based on the observed positive correlation between latency and distance travelled by data packets, these latency-based geolocation techniques constrain the estimated location of the target [18, 13, 23]. These approaches confine the region where the target is estimated to reside to within a maximum distance around each landmark. The use of a constraint for the minimum distance from the landmark, in addition to the maximum possible distance, is shown to improve the geolocation accuracy. Such positive and negative distance factors are developed based on the maximum and minimum bounds of the distance to latency relationship. However, the variability of latency measurements between Internet hosts yields a significant disparity between the positive and negative distance bounds. As a result, area to which the target is constrained using these methods is relatively large. Refining location information using additional geographical hints [18, 23] has shown to improve the geolocation accuracy. The integration of the underlying network topology information has been another method considered for improving Internet host geolocation [4].

In this paper, we present a novel geolocation algorithm, GeoWeight, which is based purely on Internet latency measurements. The GeoWeight algorithm accounts for the possible variability of distance (between the minimum and maximum possible distances) for a given latency by assigning weights to sub-regions within the region constrained by minimum and maximum bounds. The weights are assinged to sub-regions to reflect the probability that the target could be located in the respective subregion; a higher weight indicating more probable regions. The weights for the sub-regions are computed based on the Internet latency measurements for different distances as described in section 4. Latency measurements from multiple landmarks to the target result in intersect-

ing regions. GeoWieght algorithm computes a weight for an intersecting region as the sum of weights of overlapping regions enclosed in the intersection. The location of the target is chosen as the centroid of the intersection region having the highest computed weight. By assigning weights to sub-regions within the larger region, the GeoWeight algorithm is able to constrain the target location to a smaller region than that was possible from previous approaches, hence resulting in better estimation accuracies for geographical location.

In this paper we present the GeoWeight algorithm and also the technique for computing weights for different regions. We evaluate the performance of the proposed algorithm using simulated and real distance vs latency data. Through simulation, we specifically investigate two noise models for latency data, a Gamma distribution and a Lognormal distribution, to understand the impact of the noise model on the accuracy of the algorithm. We evaluate the performance of the algorithm by geolocating 60 hosts in North America. The weights for different regions in this case are computed based on large set of latency vs distance data we gathered over a month using 50 landmarks in North America using the PlatnetLab test bed. We compare our results by geolocating the same target hosts using two primarily-latency-based geolocation techniques [23, 13] and the results show that our technique outperforms both these techniques.

This paper discloses three main contributions. First, the paper develops a probability model for Internet latency. Second, it presents a novel measurement based geolocation approach, GeoWeight, for Internet host geolocation. Third, it evaluates the performance of GeoWeight and compares it with two existing measurement based approaches. The results show that GeoWeight outperforms the existing approaches in terms of *geolocation error*, distance between actual and estimated target locations.

The rest of the paper is organized as follows. Section 2 describes related work. The geolocation problem is formulated in Section 3. Internet latency data modeling and the GeoWeight algorithm are presented in Section 4. Section 5 evaluates the geolocation techniques using synthetic and measured data. Finally, conclusions are drawn and future directions are outlined in Section 6.

2 Related Work

Host Geolocation on the Internet is an important research problem that has been addressed by a number of research groups in the past. One of the intuitive approaches to host geolocation is a comprehensive IP tabulation against physical locations which can be used as a lookup table [16, 2]. However, because of the large number of available Internet hosts, such an approach will not scale. Also, a lookup table is difficult to maintain and keep up-to-date, especially, as it cannot take into consideration dynamic IP assignment.

Three techniques for geolocation were proposed in IP2Geo [18]: GeoTrack, GeoPing and GeoCluster. GeoTrack uses traceroute information from a host to the target, which contains the list of routers encountered along the path. Using location hints from the DNS names of the routers along the path, the locations of the routers are determined. Of the routers whose locations are known, the closest one to the target is selected, and its location is chosen as the target location. The accuracy of the technique depends on the distance from the target to the nearest router of known location. Next, GeoPing works on the assumption that hosts that are geographically close have similar network delays with respect to other fixed hosts. By comparing the ping times to the target from a set of landmarks or probe machines with the ping times to a set of nodes at known locations, GeoPing estimates the target location to be the same as that of the node with known location having the most similar ping values. Thus, the accuracy of GeoPing is limited by the distance to the nearest probe. The third approach, GeoCluster, is a database lookup technique which groups IP addresses to clusters based on geographical proximity. This information is combined with the user registration database from web based services such as e-mail services. This technique suffers from the general problems related to database lookup-based approaches, such as reliability, scalability and maintainability issues and also unavailability of the user registration database for public access.

Recent data-mining based approach Structon [5] is similar to GeoCluster except that it uses publicly available web pages instead of proprietary data sources in order to extract geolocation information. Structon uses a three step approach. First, extracted geolocation information from web pages are associated with their IP addresses. Then, these mapping information goes through multi-stage inference processes in order to improve the accuracy and coverage of its IP geolocation repository of different IP segments. Finally, those IP segments that are not covered in the first two steps, are mapped with the location of the access router with the help of traceroute tool. The accuracy of Structon implementation on the Internet depends heavily on the accuracy of extracted geographical mapping information. Moreover, with Structon [5], it is harder to get accuracy more than in the granularity of city level.

Constraint-Based Geolocation (CBG) [13] uses ping times from landmarks as a measure of latency. For each landmark a maximum distance bound for a given latency is derived using distance-to-ping relationships observed between landmarks. During geolocation the observed latencies from landmarks to the target are used to draw circles centered at each landmark based on the maximum distance bounds derived earlier. The target is assumed to reside in the convex region resulting from the intersection of circles, and the target location is estimated as the centroid of this convex region. This technique requires the target to be geographically well surrounded by landmarks.

Similar to CBG, Topoloy-based Geolocation (TBG) [4] computes the possible location of the target as a convex region. In TBG, the maximum distance bound is obtained based on the maximum transmission speed of packets in fibre which gives a conservative estimate of the possible region. This region is further refined using inter-router latencies along the path from the target to the landmark, obtained from the traceroute command. The final target location is obtained through a global optimization that minimizes average position error for the target and the routers.

A more recently proposed measurement-based technique for geolocation is Octant [23]. In contrast to other constraint based approaches that only limit the area where the target may be located, Octant also identifies areas where the target may not be located based on observed latencies (referred to as negative constraint). Octant expresses such information by considering two circles corresponding to the maximum and minimum distances from each landmark to the target which constrains the possible geographical area where the target may be located. Each landmark fits a convex hull to all of its delay-to-distance data points with other landmarks. Upper and lower facets of the convex hull correspond to the maximum and minimum distance bounds. Different weights are

assigned to different geographical areas based on the number of intersections (higher weights assigned to larger numbers of intersections). The final estimated region is the union of all regions, where the weight exceeds a desired weight or the region size exceeds a selected threshold. A Monte-Carlo algorithm is applied to pick the best single point location from the final estimated regions. These estimated regions in Octant often end up being disconnected parts. In contrast, it is highly unlikely with GeoWeight. As in GeoWeight the maximum (positive) and minimum (negative) distance bounds are divided into different weighted regions. Octant uses geographical and demographical constraints to improve the localization accuracy beyond its measurement-only solution.

In addition to the above cited references [10, 17, 24, 12] also discuss Internet host geolocation. Other relevant research which includes geographic properties of routing [21], delay prediction or distance estimation between Internet nodes [6, 22, 15, 11, 19] and exploring nearby servers [14] have also contributed to the area of host geolocation.

GeoWeight differs from other measurement based approaches because it uses a weighted model for latency-to-distance measurements to estimate the target location.

3 Problem Formulation

This section presents the problem statement for GeoWeight.

3.1 Problem Statement

The problem considered here is the geolocation of a target H. Let us denote the unknown position P_0 of the target in terms of its latitude and longitude (lat_0, lon_0).

Suppose that $\{L_1, L_2, \ldots, L_N\}$ be a set of N landmarks. Let (lat_i, lon_i) be the latitude and longitude of the ith landmark L_i. We carry out geolocation using latency measurements from the N landmarks to the target. Let $\mathbf{t}_i = \{t_{i,j}\}_{j=1}^{n_i}$ be the set of n_i latency measurements from landmark i to the target. We denote the cumulative set of all measurements from landmarks to the host by: $\mathbf{t}_{1:N} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N\}$.

Our goal is to estimate the location P_0 of target H using measurements $\mathbf{t}_{1:N}$. We denote the estimated location of the target by $\hat{P}_0$.

Then the *geolocation error* ϵ is defined as,

$$\epsilon = dist(P_0, \hat{P}_0) \tag{1}$$

where $dist(P_x, P_y)$ represents the geographical distance between position P_x and P_y.

3.2 The Latency Model

Considering that Internet data packets in the majority of the cases travel through optical fibres, the minimum latency between two nodes that are a distance d apart can be given by,

$$t_{min}(d) = d/c_{fibre} \tag{2}$$

. Here c_{fibre} is the maximum transmission speed of data through the fibre, which is approximately 2/3 the speed of light [13]. As we show later, the latencies observed in real world Internet traffic are significantly higher than this lower bound due to factors such as router congestion. Thus, the observed latencies are modeled as,

$$t(d) = t_{min}(d) + E(d) \tag{3}$$

where $t(d)$ is the observed latency for distance d and $E(d)$ is a noise term that accounts for network delays in the real measurement.

4 The GeoWeight Algorithm

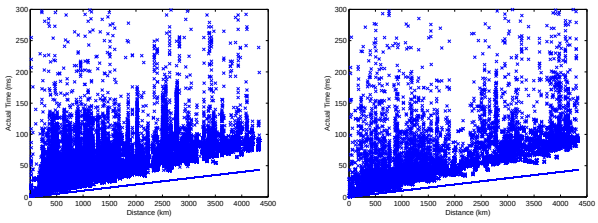
This section presents the GeoWeight algorithm.

4.1 Initial Observations

Figure 1(a) shows a plot of the distance vs latency (approximately 150,000 data points) gathered on the PlanetLab test bed using 50 PlanetLab nodes in North America as landmarks. The Internet Control Message Protocol (ICMP) [3] ping delay between landmarks was used as the measure of latency. More details of the experimental setup is described in section 5.2.1. The solid line below the data points in figure 1(a) shows $t_{min}(d)$ given by equation 2. Figure 2 shows the histogram of observed distances for a given latency range (90.05 ms-100.00 ms). In order to ascertain the observation of our PlanetLab dataset we also analyzed the dataset collected by iPlane [1] during same period. This dataset is based on latency measurements, shown in figure 1(b), between their 68 landmarks which similar to our landmarks are spread around North America.

Following are five characteristics observed from this data (figure 1(a) and 1(b)):

- The minimum latency observed is higher than the theoretical minimum given by the equation 2.
- There is a positive correlation between latency and distance.
- A simple linear or non-linear relationship is not apparent in the data set - the data in noisy as described by equation 3.
- Although an upper bound on distance for a given latency is apparent, a lower bound is not apparent. This is the case even for the data analyzed on a per landmark basis.
- For a given latency (or a latency range), some distances are more probable than other distances (Figure 2).



(a) (b)

Figure 1: Distance-latency relationship (a)between 50 PlanetLab Landmarks (b)between iPlane dataset landmarks. The straight line below the data points shows delay to distance relationship according to equation 2

4.2 Existing Techniques

The current latency based approaches for target geolocation attempt to take into consideration the first four of the above observed characteristics of the distance to latency relationship.

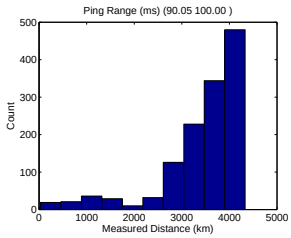


Figure 2: Histogram of distances for latency range 90.05-100.00 ms in the PlanetLab dataset

The CBG [13] technique uses the maximum distance bound in order to constrain the target location. Based on the observed latency from a landmark, the target location is constrained to the circular region around the landmark based on the upper bound of distance. Considering latency measures from multiple landmarks, the region of the target is considered as the convex region with maximum number of intersecting regions. The centroid of this convex region is estimated as target location. Figure 3 shows an example of the CBG approach for the case of a target being geolocated using three landmarks. As the figure shows, the CBG approach rightfully constrained the target inside the convex region based on the maximum distance bound. However, the drawback of this approach is the relatively large area the target is constrained to, around the landmark, due to considering only the maximum distance bound. This results in a relatively large final target region, hence potentially large geolocation errors.

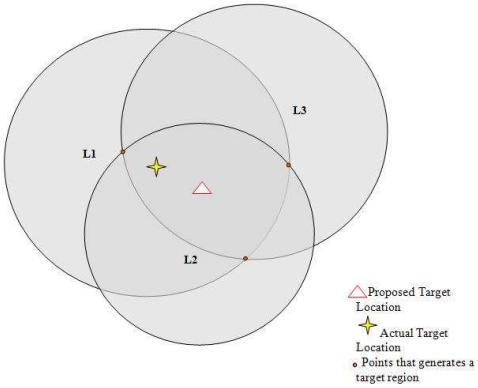


Figure 3: CBG Example

The Octant [23] approach improves the CBG approach by considering negative constraints. This technique defines the maximum-bound of distance as a positive constraint, within which the node must lie and a negative constraint, which indicates a minimum distance from the landmark where the target is constrained not to be present. Figure 4 shows an example of Octant approach for the case where a tagert is geolocated using three landmarks. Compared to the CBG approach, Octant reduces the possible target region size with the help of negative constraints. However, the latency vs distance measures we gathered show that the lower bound of distance for a given ping time is not apparent. As a result, constrained region produced by this technique could still be large resulting in high geolocation errors.

4.3 GeoWeight Approach

In addition to the first four characteristics of the distance to latency relationship identified in section 4.1,

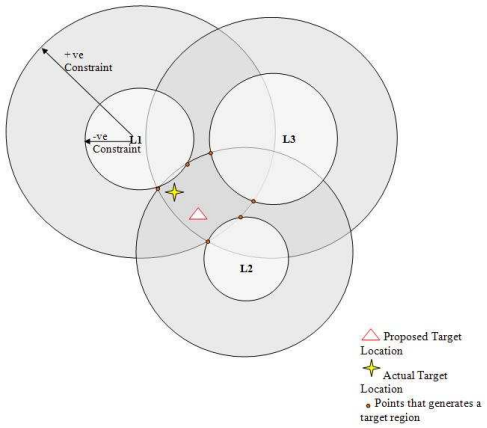


Figure 4: Octant Example

the GeoWeight algorithm takes into consideration the fifth characteristic of the distance-latency relationship; for a given latency, some distances are more probable than other distances. The GeoWeight algorithm uses this characteristic to constrain the possible region of the target to a smaller region than that was possible in the previous approaches as we describe below.

Let t_x be an observed latency from an arbitrary landmark. Based on the distance-latency relationship let d_x^{min} and d_x^{max} be the minimum and maximum possible distances for t_x. Consider the distance range from d_x^{min} to d_x^{max} is divided to $N_{x,d}$ number of equal sized distance bins. The j-th bin, (j = 1, 2,....$N_{x,d}$), covers the distance range from $d_{x,j}^{min}$ to $d_{x,j}^{max}$ given by,

$$d_{x,j}^{min} = d_x^{min} + (j-1)(d_x^{max} - d_x^{min})/N_{x,d}$$

$$d_{x,j}^{max} = d_x^{min} + j(d_x^{max} - d_x^{min})/N_{x,d}$$

Let $w_{x,j}$ be the weight for the j-th distance bin corresponding to t_x. The weight $w_{x,j}$ represents the probability of the distance being in the range $d_{x,j}^{min}$ and $d_{x,j}^{max}$ for the observed latency t_x. The details of weight computation will be described in section 4.4.

For a given latency t_x, the GeoWeight algorithm considers $N_{x,d}$ number of regions around the landmark, with the j-th region having distance bounds $[d_{x,j}^{min}, d_{x,j}^{max}]$ and a weight of $w_{x,j}$. The latency measurements from multiple landmarks will result in intersecting regions, with different numbers of overlapping regions in each intersection region. The final weight for each intersection region is computed as the sum of weights of intersecting regions. The region of highest weight is considered as the constrained region of the target and the centroid of the region is estimated as the target location.

We simplify the generic algorithm presented above by considering the minimum distance, maximum distance and the number of distance bins to be the same for any latency, i.e,

$$d_x^{min} = D_{min}$$

$$d_x^{max} = D_{max}$$

$$N_{x,d} = N_d$$

where N_d is the number of distance bins for any latency under consideration and D_{min} and D_{max} are the minimum and maximum possible distances for the geolocation scenario. For example, in section 5, where we evaluate our algorithm, the considered

Table 1: An example weight table for GeoWeight

Ping Time	0-250(km)	250-500 (km)	500-750 (km)	750-1000 (km)
100	0	0	0.2	0.8
35	0.2	0.6	0.2	0
15	0.7	0.2	0.1	0

D_{min} and D_{max} for the geolocation scenario are set such that it covers the whole of the North-American region. The only implication of the above simplification is some distance bins having a weight of 0 due to these distance ranges not being probable for the particular latency.

Figure 5 illustrates an example of the GeoWeight algorithm for a geolocation scenario with latency measurements from three landmarks. In this example the observed latency measures from the three landmarks L1, L2 and L3 are 100, 35 and 15 ms respectively. Table 1 shows the computed weights for the three latencies for different distance regions considering four ($N_d = 4$) equidistant bins in the distance range 0-1000 km (in this example, $D_{min} = 0$, $D_{max} = 1000$).

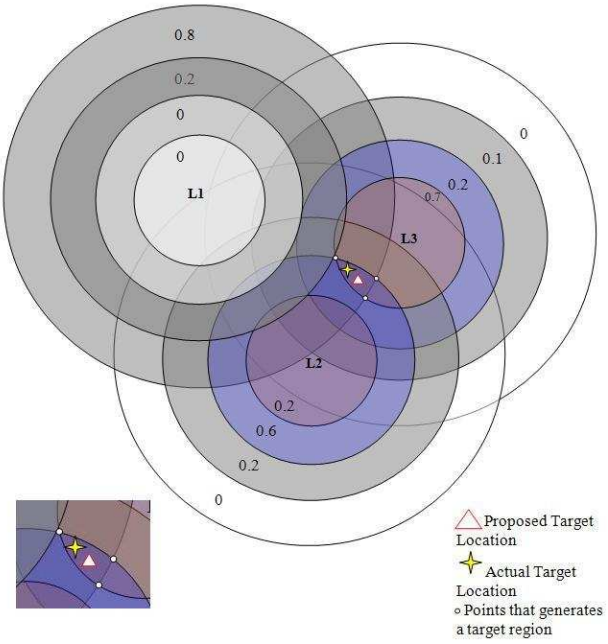


Figure 5: GeoWeight Example

Figure 5 shows the four regions ($N_d = 4$) with different weights around each landmark in form of circles. These circles overlap with each other and the weight of each intersection region is computed as the sum of weights of the overlapping circles in the intersection region. For clarity, figure 5 does not show weights of all regions.

In this example, the region of maximum weight has a weight of 2.1 (0.8 + 0.6 + 0.7) and the final target location is selected at the centroid of this region as shown in figure 5.

4.4 Weight Computation

This section describes how the weights for the different distance regions are computed. As mentioned before, the weight for a distance range for a given latency is the probability of the distance being in the range for the given latency.

Consider a latency vs distance data set gathered from Internet measurements covering a distance range D_{min} to D_{max}. Let T_{min} and T_{max} be the minimum and maximum observed latencies. Consider this distance range and the time range divided to N_d and N_t equidistant bins respectively. Let c_{ij} be the number of data point corresponding to the i-th time and j-th distance bin.

Table 2: Weight computation example: the number of observed measurements for different time and distance ranges are shown in the table

	(0-500)km	(500-1000)km	(1000-1500)km	(1500-2000)km
(0-10)ms	100	25	0	0
(10-20)ms	15	120	35	0
(20-30)ms	12	52	95	12
(30-40)ms	5	23	126	32
(40-50)ms	2	10	24	68
(50-60)ms	0	5	12	128
(60-70)ms	0	12	21	45
Total	134	247	313	285

Table 2 shows an example of delay and distance bins. In this example, $T_{min} = 0$, $T_{max} = 70$, $D_{min} = 0$, $D_{max} = 2000$, $N_d = 4$ and $N_t = 7$.

Each row represents the distance bins corresponding to a single time bin. Each column represents the time bins corresponding to a single distance bin. Each cell of the table shows the number of data points within a given time-distance bin.

Since the latency measures are collected for specific distances (i.e. inter landmark distances), even if the same number of latency measurements is gathered for each distance, the total number of distances represented in each distance bin will vary between distance bins as shown in table 2. Therefore, the first step in computing the weights is the normalization across distance bins by dividing the number in each distance-latency cell by the total number of measurements for the particular distance bin. The normalized distance-latency $NR_{i,j}$ are given by,

$$NR_{i,j} = c_{i,j} / \sum_{i=1}^{N_t} c_{i,j} \quad (4)$$

The final weight for each cells is computed by normalizing across the latency bin, given by,

$$w_{i,j} = NR_{i,j} / \sum_{j=1}^{N_d} NR_{i,j} \quad (5)$$

where $w_{i,j}$ is the weight of ith latency bin of jth distance bin which is the probability of the distance region $[d_j^{min}, d_j^{max}]$ given the observed latency is in the range $[t_i^{min}, t_j^{max}]$.

In the example shown in table 2, considering the bin $i = 1$ and $j = 1$, $NR_{1,1} = 100/134$ (0.74), $NR_{1,2} = 25/247$ (0.10), $NR_{1,3} = 0/313$ (0) and $NR_{1,4} = 0/285$ (0). The sum of normalized weights S_1, (0.74 + 0.10 + 0 + 0) = 0.84. Therefore $w_{1,1} = NR_{1,1}/S_1 = 0.88$, $w_{1,2} = NR_{1,2}/S_1 = 0.11$, $w_{1,3} = NR_{1,3}/S_1 = 0$, $w_{1,4} = NR_{1,4}/S_1 = 0$.

5 Evaluation

This section presents the results of the experiments we conducted to evaluate our algorithm. We evaluated GeoWeight using simulated data as well as real Internet data. We also compared GeoWeight with two existing techniques; Octant and CBG. The results are presented in sections 5.1 and 5.2 respectively.

5.1 Evaluation based on Simulated Data

We conducted experiments using latency measurements simulated based on the latency model presented in section 3 using two different probability

models for noise. The aim of these experiments was to:

- Determine the optimum values of N_d and N_t for the algorithm.
- Evaluate the algorithm for known noise models.

The results of these experiments are presented in the following sections.

5.1.1 Experiment 1: Determining N_d and N_t

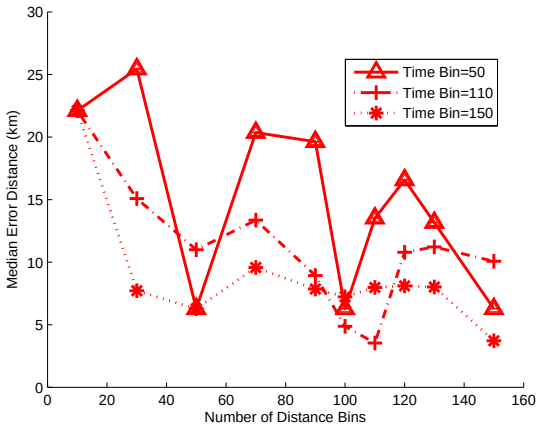


Figure 6: Performance of GeoWeight with different time and distance bins combination in noise free case

The GeoWeight algorithm considers finite size bins for distance and time for weight computation. As a result, when latency measures from multiple landmarks is considered, even for the noise free case, GeoWeight produces region of maximum weight as opposed to a single point. Therefore, the GeoWeight algorithm will not result in a zero geolocation error even for the noise free case, unlike distance based triangulisation that would give a zero error. The accuracy of the GeoWeight algorithm will depend on bin size. Smaller bin sizes will result in lower errors provided the weight for the bin can be computed accurately. However, the accuracy of weight computation will be limited by the number of available measurements for a distance-time bin. Therefore the resolution of the distance bins will be determined by the number of distances in the data set. Similarly the number of data points for a distance bin will determine the resolution of the time bins. Thus, the goal of this experiment was to investigate the best possible bin sizes (determined by the number of time bins(N_t) and number of distance bins (N_d)) to be used for computing the weights.

In this experiment, we use simulated latency data generated for the 1200 inter landmark distances in our data set with 1000 data points for each distance. We consider the noise free case (equation 2) where the distance to latency relationship is linear. We estimate the geolocation error (ϵ) by varying N_t and N_d. We geolocate 60 targets using 50 landmarks. The location of the landmarks and the targets are the same as our real landmarks and targets locations as described in section 5.2.1.

Figure 6 shows the median geolocation error results of three distinct values of N_t for N_d in the range of 10-150. The figure shows results of only three N_t for simplicity. As the figure shows [(N_t, N_d)=(110, 110)] combination resulted in the lowest median geolocation error distance . Therefore, [(N_t, N_d)=(110, 110)] is used for weight computation in the subsequent experiments.

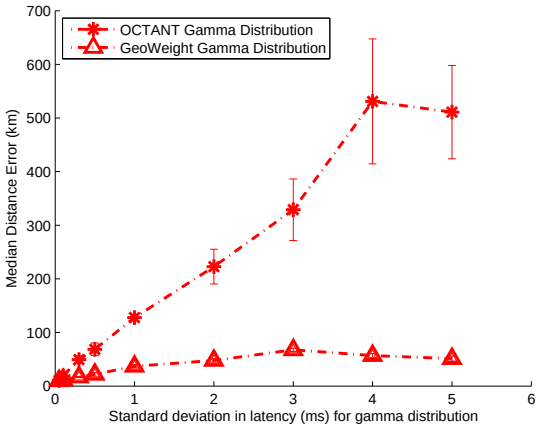


Figure 7: Median geolocation error as a function of variance for GeoWeight and Octant with Gamma noise distribution with mean = 5

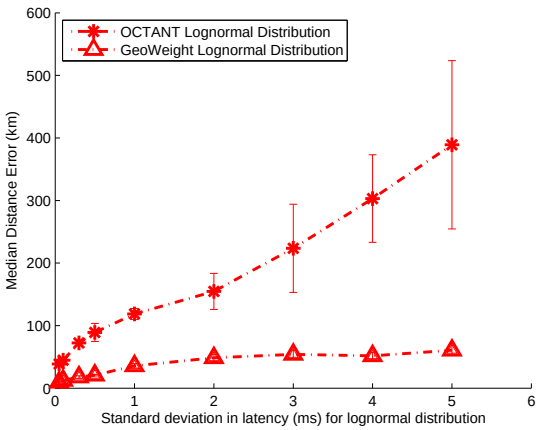


Figure 8: Median geolocation error as a function of variance for GeoWeight and Octant with Lognormal noise distribution with mean = 5

5.1.2 Experiment 2: Evaluation of GeoWeight

The goal of this experiment was to investigate the impact of different noise models on the geolocation error (ϵ). We also compare our algorithm with Octant's measurement-only solution for the different noise models.

In this experiment we investigated two different noise models: lognormal and gamma distribution. These two distributions were chosen because of their skewed and non-negative properties [8], similar to our real data set. Moreover, previous study [9] found lognormal characteristics in the Internet latency distribution. However, we acknowledge that these noise models do not accurately models the observed latencies. The experiments are to identify the behavior of the algorithm and understand the theoretical limits of accuracy of our algorithm.

The probability distribution function of the Gamma distribution is given by:

$$p(t|a,b) = \frac{1}{b^a \Gamma(a)} t^{a-1} \exp^{\frac{-t}{b}}$$

where a, b are the parameters of the distribution. The mean m and the variance v are given by,

$$m = ab \quad (6)$$
$$v = ab^2 \quad (7)$$

The probability distribution function of the Lognormal distribution is given by:

$$p(t|\mu,\sigma) = \frac{1}{\sigma t\sqrt{2\pi}} \exp \frac{-(\ln(t)-\mu)^2}{2\sigma^2}$$

The mean m and the variance v are given by,

$$m = e^{\mu+\sigma^2/2} \quad (8)$$
$$v = (e^{\sigma^2}-1)e^{2\mu+\sigma^2} \quad (9)$$

where μ, σ are the mean and the standard deviation of the corresponding normal distribution.

We generated the weight matrix using simulated distance-latency data for the 1200 inter landmark distances using the noise model. Similar to experiment 1, we geolocated 60 targets using 50 landmarks to evaluate the performance of GeoWeight. The latency measures from the landmark to the target were also generated using the same noise model. We computed the median geolocation error by varying the mean and the variance of noise. We also computed the geolocation error of the Octant algorithm for these noise models. The measurement-only step of Octant algorithm was implemented using the algorithm details provided in [23].

Figure 7 and 8 show the median gelocation error for GeoWeight and Octant for different noise variances for Gamma and Lognormal noise distributions respectively. The mean value of noise is chosen to be 5 in each case. We have run the experiment for different values of mean but did not observe significant differences in the results for GeoWeight and Octant. Thus, for clarity figure 7 and 8 only show the case of $m = 5$.

For both noise models, it is evident that the median geolocation error increases with the increase in noise variance as expected. The rate of increase is higher for Octant than GeoWeight which shows that the geolocation error of Octant is more sensitive to noise than GeoWeight. This is due to the increase in the difference between the maximum and minimum distance bounds for higher noise variances. This results in a large bounding region for Octant whereas GeoWeight accommodates this difference by dividing the region to small regions of different weights.

5.1.3 Experiment 3: The impact of the number of landmarks

In this experiment we investigate the impact of the number of landmarks on the geolocation error distance ϵ. In each experiment we select a random subset of the 50 landmarks, and we use this subset to geolocate the 60 targets.

Figure 9 shows median geolocation error as a function of the number of landmarks for Gamma and Lognormal noise distributions. The noise mean and variance were chosen to be 4 and 16 respectively. It can be seen that the geolocation error decrease with the increase in the number of landmarks. This is because higher number of landmarks allow to better constrain the target region since the intersection regions end up smaller.

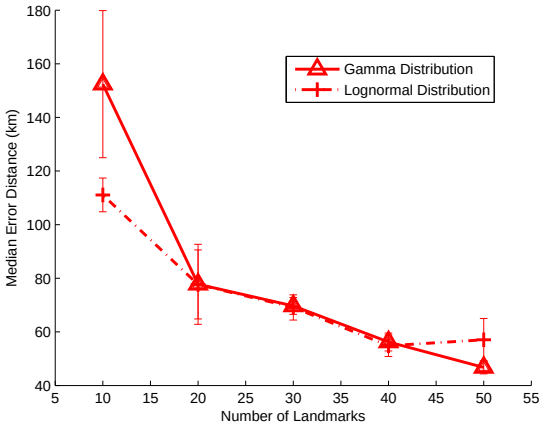


Figure 9: Median geolocation error as a function of the number of landmarks for Gamma and Lognormal noise distributions with $m = 4$ and $v = 16$.

5.2 Evaluation on Real Data

This section evaluates GeoWeight by geolocating real targets on the Internet.

5.2.1 Experimental Setup

Our experimentation was carried out on PlanetLab (`www.planet-lab.org`), which is an experimental test bed on the Internet. We collected latency data, generated via ping tool, from PlanetLab nodes consisting of 50 landmarks in North America. The location of the landmarks is shown in figure 10. North America covers large geographical area and possesses substantial number of users, hosts and network connectivity of the Internet. Thus, we believe the methodology that we developed in this paper is notable even though we limited our scope to North America. However, the delay-distance relationship may vary based on different geographical location.

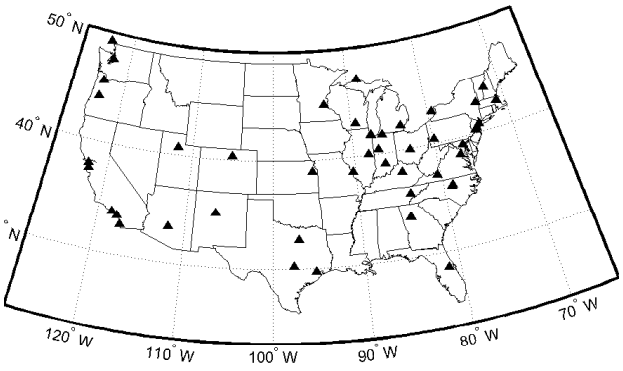


Figure 10: Location of the chosen landmarks

The latency data between landmarks were collected over a period of one month, from September 23, 2008 to October 25, 2008. Over this period, we executed a script, on each landmark, which generated ping commands originating from this landmark to every other landmark. At a given time, the originating landmark performed a three data packet ping to a selected destination landmark followed by a two

minute pause. Then the process continued with a new destination. After cycling through all destination landmarks the process was repeated. From the three observed ping times, the minimum ping time was chosen as the measure of latency for modelling purposes.

The full dataset we gathered consists of approximately 150,000 distance-latency measurements. Although we used 50 landmarks, equal number of measurements were not available for each landmark due to certain ping commands not successfully completing from these PlanetLab nodes during the measurement collection period. The inter-landmark distance in the data covers the range 0.5 km - 4331 km. This data set was used for computing the weights.

During the same period, we also gathered latency data from 50 landmarks as described above to 60 targets in North America. None of our targets was in the same domain as the landmark. This data set was used for geolocating the targets.

5.2.2 Experimental Results

We first computed the weights w_{ij} for the GeoWeight algorithm using the latency measurements between the 50 landmarks. We then geolocated the 60 targets using latency measurements to these targets from the 50 landmarks by choosing one random ping time from each landmark. We geolocated each target four times each time selecting a single random ping time from the data set for each landmark ($n_i = 1$).

Figure 11 shows the cumulative distribution of the mean geolocation error for GeoWeight, Octant and CBG. First of all, since Octant's implementation of the published algorithm was not available to us, we implemented the latency measurement based geolocation step of the Octant algorithm. We did not include the additional optimizations of the Octant algorithm since our goal is to compare with the measurement-only geolocation approach. The maximum and minimum distance bounds in this case were computed, per landmark basis, using the latency data gathered between PlanetLab landmarks. We geolocated the same 60 targets with GeoWeight and Octant using our latency measurements. Figure 11 shows the cumulative geolocation error distribution of GeoWeight and Octant (our implementation) in solid and dotted line respectively. The median geolocation error for GeoWeight is 44 km and for Octant is 456 km.

Octant error is significantly higher compared to the published results in [23]. This we believe is due to two reasons. First, Octant approach is based on minimum and a maximum distance bounds for a given latency. However, with both the PlanetLab (figure 1(a)) and iPlane (figure 1(b)) dataset we have not seen a clear minimum distance bound of distance-latency relationship, even for per landmark basis. The lack of a minimum distance bound is further confirmed in [7] based on their dataset. The lack of a minimum distance bound results in a larger constraint region, resulting in higher error. Second, in the published work [23] Octant used different optimization techniques. Since our goal is to compare the geolocation approaches based purely on latency measurements we did not include such optimization techniques in either algorithm. This gives a fair comparison between the approaches. However, the additional optimizations used by Octant can also be incorporated into our algorithm.

We also geolocated the same 60 targets by Octant and CBG using the geolocation service provided by the authors of Octant [23]. This service is available at: `http://www.cs.cornell.edu/~bwong/octant/query.html`. Median geolocation errors of Octant and CBG by the service provided by the authors of [23] are 216 km and 506 km respectively. Figure 11 shows the cumulative geolocation error distribution of Octant and CBG obtained from this service in dash and dash-dotted line respectively. We geolocated each target three times and the geolocation error shown in figure 11 is the minimum of the three values. This geolocation service uses supplementary constraints such as geographical and demographical hints in addition to latency measurements to refine its estimates whereas our implementation of Octant does not use any such optimization technique. We believe that this is the reason for the observed differences of geolocation error between our implementation of Octant and the implementation provided by the authors. Extra geographical and demographical hints significantly improved Octant's accuracy and this observation is aligned with the description in [23]. The median and mean estimation errors for the three approaches are listed in table 3.

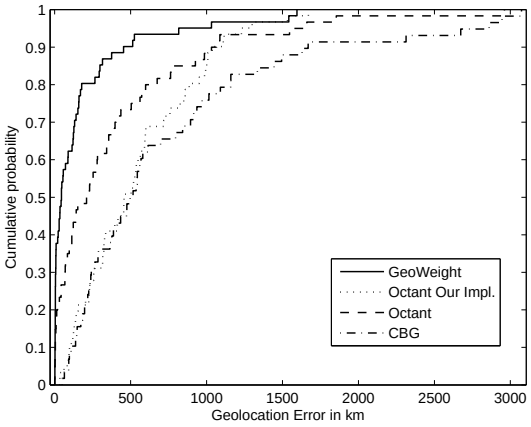


Figure 11: Cumulative distribution of geolocation error distance for GeoWeight, Octant and CBG

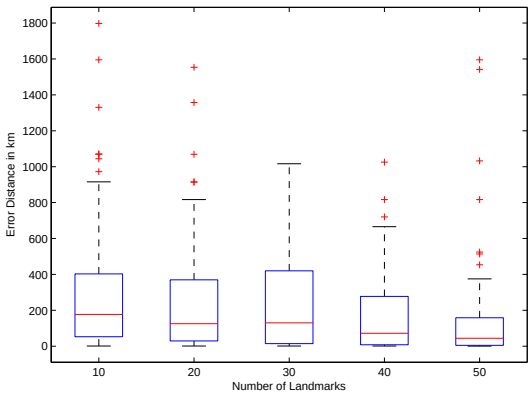


Figure 12: GeoWeight's performance with real data with different number of landmarks

We then investigated the impact of number of landmarks on the performance of GeoWeight. This experiment used a randomly chosen subset of landmarks which were used to geolocate the 60 targets. Figure 12 shows the box plot of geolocation error as a function of the number of landmarks. Figure 12 shows the accuracy increases with increase in the number of landmarks. However, it is to be noted that even with a smaller number of landmarks GeoWeight

Table 3: The mean and median geolocation error (rounded to the nearest kilometer) for GeoWeight, Octant and CBG approaches

	GeoWeight	Octant Our Imp.	Octant	CBG
Mean (km)	170	541	396	749
Median (km)	44	456	216	506

still performs better than existing approaches (Table 3).

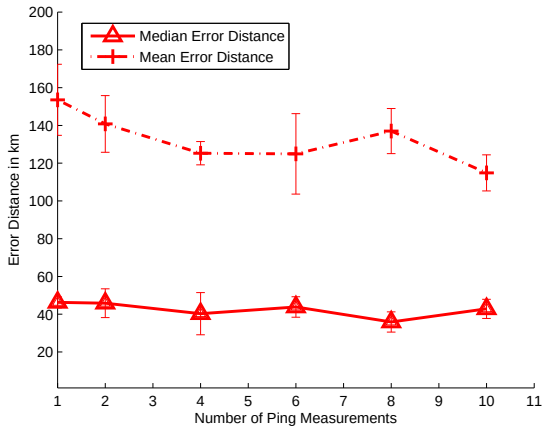


Figure 13: GeoWeight's Performance with multiple ping times

Finally, we investigated the impact of number of measurements, n_i, between each landmark and target pair on the geolocalization error of GeoWeight.

Figure 13 shows GeoWeight's performance for different values of n_i. The mean and median geolocation error for 60 targets using 50 landmarks is shown in figure 13. The experiment was repeated four times and the error-bars display the standard deviation of these four results.

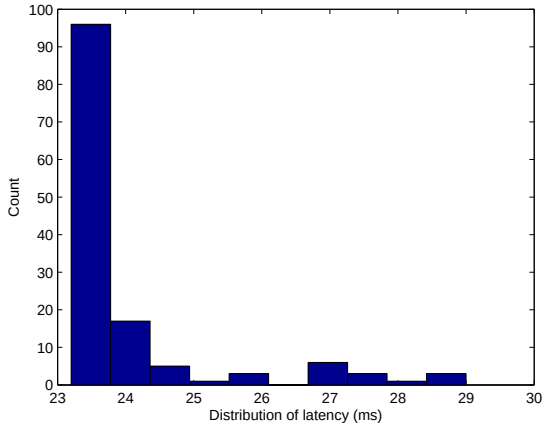


Figure 14: Distribution of latency between Boston University and University of Chicago

It is observed that there is no significant difference in the observed error based on n_i. In our dataset, we have observed that for a given landmark-target pair most of the ping times observed are similar with some outliers as shown in the histogram in figure 14, which could explain why n_i does not have an impact on the geolocation error.

6 Conclusions

In this paper, we have presented GeoWeight, a novel latency-based geolocation algorithm. GeoWeight is based on a probability model for Internet latency computed from observed latencies for different distances. We validated GeoWeight with simulated and real Internet data using PlanetLab as a test bed. The results show that GeoWeight outperforms existing approaches such as CBG and Octant. GeoWeight was able to geolocate 60 targets in North America using 50 landmarks with a median geolocation error of approximately 44 km. This geolocation approach was based purely on latency measurements as opposed to the existing techniques that supplement the latency data with geographical constraints in order to increase their accuracy.

As our future work we are investigating better probability distributions for Internet latency data which can capture the behavior of noise in latency on a per landmark basis. One initial investigations have focussed on the Levy distribution which have shown promising results.

References

[1] iPlane: Latency Measurements from University of Washington. *http://www.mcs.anl.gov/ olson/IPtoLL.html*.

[2] IP to Latitude/Longitude. *http://www.mcs.anl.gov/ olson/IPtoLL.html*.

[3] B.A.Forouzan. *Data Communications And Netoworking*. Tata McGraw-Hill Publishing Company Limited, 2000.

[4] Basset, E., John, P., Anderson, T., Krisnamurthy, A., Chawathe, Y., and Wetherall, D. Towards IP Gelolocation Using Delay and Topology Measurements. *In the Proceedings of IMC 06.* (2006).

[5] C.Guo, Y.Liu, W. H. Q. Y. Mining the Web and the Internet for Accurate IP Address Geolocations. *In Proceedings of INFOCOM 2009. The 28th Conference on Computer Communications. IEEE* (2009).

[6] Dabek, F., Cox, R., Kaashoek, F., and Morris, R. Vivaldi: A Decentralized Network Coordinate System. *In the proceedings of the SIGCOMM 2004* (2004).

[7] D.Li, J.Chen, C. Y. J. Z. Y. IP-Geolocation Mapping for Involving Moderately-Connected Internet Regions. *Project participation from Microsoft Research: http://research.microsoft.com/en-us/people/danil/* (2009).

[8] E.Limpert, W.A.Stahel, M. Log-normal Distributions across the Sciences: Keys and CLues. *In the Proceedings of BioScience, Vol. 51 No. 5 pp. 341-352.* (2001).

[9] F.Melakessou, U.Sorger, Z. On The Road Towards The Comprehension Of The Internet Traffic Behavior: Simulation And Analysis Of An End-To-End Connection With NS-2. *In the Proceedings of the 2007 spring simulaiton multiconference - Volume 1.* (2007).

[10] Fossen, E., and Arnes, A. Forensic Geolocation of Internet Addresses using Network Measurements. *Nordsec 2005, 10th Nordic Workshop on Secure IT-systems* (2005).

[11] Francis, P., Jamin, S., Jin, C., Jin, Y., Raz, D., Shavitt, Y., and Zhang, L. IDMaps: a global Internet host distance estimation service. *IEEE/ACM Transactions on Networkding, Volume 9, Issue 5, Oct. 2001 Page(s):525 - 540* (2001).

[12] Gueye, B., Uhlig, S., Ziviani, A., and Fdida, S. *Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems.* Springer Berlin / Heidelberg, 2006, ch. Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts, pp. 319–330.

[13] Gueye, B., Ziviani, A., Crovella, M., and Fdida, S. Constraint-Based Geolocation of Internet Hosts. *Networking,IEEE/ACM Transactions 14*, 6 (2006), 1219–1232.

[14] Guyton, J. D., and Schwartz, M. F. Locating nearby copies of replicated Internet Servers. *In the Proceedings of ACM SIGCOMM 95, Cambridge, MA. U.S.A.* (1995).

[15] Madhyatha, H. V., Anderson, T., Krishnamurthy, A., Spring, N., and Venkataramani, A. A Structural Approach to Latency Prediction. *In the Proceedings of IMC 06, Rio de Janeiro, Brazil* (2006).

[16] Moore, D., Periakaruppan, R., and Donohoe, J. Where in the World is netgeo.caida.org?. *Presented in INET2000 Poster (Yokohama, Japan, July 2000)* (2000).

[17] Muir, J., and Oorschot, P. C. Internet Geolocation and Evasion. *Technical Report TR-06-05, School of Computer Science, Carleton University.* (2006).

[18] Padmanabhan, V. N., and Subramanian., L. An investigation of geographic mapping techniques for internet hosts. *In the Proceedings of ACM SIGCOMM Computer Communication Review* (2001).

[19] Pietzuch, P., Ledlie, J., Mitzenmacher, M., and Seltzer, M. Network-Aware Overlays with Network Coordinates. *In the Proceedings of 26th IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW06* (2006).

[20] Roxin, A.; Gaber, J. W. M. N.-S.-M. A. Survey of Wireless Geolocation Techniques. *In the proceeding of the IEEE Globecom Workshops, 2007.* (2007).

[21] S, L., N., V., and H., R. Geographic Properties of Internet Routing. *Proceedings of the General Track: 2002 USENIX Annual Technical Conference* (2002).

[22] Wang, B., Slivkins, A., and Sirer, E. M. Meridian: A Lightweight Network Location Service without Virtual Coordinates. *In the Proceedings of ACM SIGCOMM 05, Philadelphia, Pennsylvania, U.S.A.* (2005).

[23] Wong, B., Stoyanov, I., and Sirer, E. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. *In Proceedings of Symposium on Networked System Design and Implementation, Cambridge, Massachusetts* (2007).

[24] Ziviani, A., Fdida, S., de Rezende, J. F., and Duarte, O. C. M. B. Improving the accuracy of measurement-based geographic location of internet hosts. *Computer Networks and ISDN Systems archive, Volume 47 , Issue 4 (March 2005)* (2005).