Statistical Geolocation of Internet Hosts

Inja Youn
Dept. of Computer Science
George Mason University
Fairfax, Virginia 22030 USA
Email: iyoun@gmu.edu

Brian L. Mark
Dept. of Electrical & Comp. Eng.
George Mason University
Fairfax, Virginia 22030 USA
Email: bmark@gmu.edu

Dana Richards
Dept. of Computer Science
George Mason University
Fairfax, Virginia 22030 USA
Email: richards@cs.gmu.edu

***Abstract*—Automated geolocation of IP addresses has important applications to targeted delivery of local news, advertising and other content over the Internet. Previous measurement-based approaches to geolocation employ active probing to measure delays among a set of landmark nodes with known locations. The location of a target IP address can be approximated by that of the nearest landmark, as determined by the delay measurements. To improve geolocation accuracy, a variation of this approach uses multilateration with geographic distance constraints to obtain a continuous location space rather than the discrete set of landmark locations. Since the previous approaches are fundamentally deterministic, they can only provide relatively loose bounds on the true location of an IP address. We develop a statistical geolocation scheme based on applying kernel density estimation to delay measurements among a set of landmarks. An estimate of the target location is then obtained by maximizing the likelihood of the distances from the target to the landmarks, given the measured delays. This is achieved by an algorithm which combines gradient ascent and force-directed methods. We present experimental results on PlanetLab to demonstrate the superior accuracy of the proposed geolocation scheme compared to previous methods.**

***Index Terms*—Geolocation, delay measurements, kernel density estimation, maximum likelihood estimation, force-directed algorithm.**

I. Introduction

Geolocation of IP addresses has important applications to targeted delivery of local news, advertising and other content over the Internet. One approach to IP geolocation is to maintain a large distributed database containing mappings of IP addresses to geographical locations. Many sites rely on more or less accurate databases to determine the location of a customer for various reasons:

- determining regional distribution of the clients, local news delivery, targeted advertising, restriction of content delivery based on regional policies, etc.;
- prevention/reduction of Internet frauds such as credit card fraud, identity theft, spam and phishing;
- application to intrusion detection.

Since such databases are often proprietary and manually updated, their consistency and accuracy are questionable at best. In addition, with the advent and adoption of IPv6, such databases become more difficult to update and maintain. Moreover, a large IP geolocation database cannot adapt easily to the frequent location changes of mobile targets. As an alternative approach, RFC 1876 proposes to incorporate geographical information into DNS records (DNS LOC). However, the implementation of this approach is not currently widespread, since it requires changes in the DNS records.

Recently there have been efforts to automate the geolocation of IP addresses. Padmanabhan and Subramanian [1] have investigated three techniques: inferring the geographic location of an Internet host based on the DNS name of the host or another nearby node (GeoTrack); clustering the IP address space into likely collocated clusters (GeoCluster); and pinging the host, with geolocation of the IP address performed by correlating ping delays (GeoPing). The latter approach employs active probes to measure delays among a set of *landmark* nodes with known locations. Such delay measurements can be performed by a distributed network of servers. Such a network of servers can be self-calibrating and potentially able to detect when a target IP address has changed its geographical coordinates significantly. However, inferring geographical location from Internet delay measurements may result in large errors due to the nonlinear relationship between geographical distance and "Internet" distance as determined from delay measurements.

Given delay measurements among a set of landmark nodes with known locations, the location of a target IP address can be approximated by that of the nearest landmark. To improve geolocation accuracy, a variation of this approach uses multilateration with geographic distance constraints to obtain a continuous location space rather than the discrete set of landmark locations. Gueye *et al.* [2] improved upon GeoPing using an idea borrowed from sensor networks. Their Constraint-Based Geolocation (CBG) algorithm uses a multilateration algorithm to determine the probable location of the targets. Katz-Bassett *et al.* [3] proposed Topology Based Geolocation (TBG), which finds hosts along Internet paths using the *traceroute* utility and geolocates hosts simultaneously using CBG. All of these methods are based on deterministic algorithms, which can have unacceptable geolocation errors of more than 1000 km.

In this paper, we develop a statistical geolocation scheme based on applying kernel density estimation to delay measurements obtained among a set of landmarks. An estimate of the target location is then obtained by maximizing the likelihood of the distances from the target to the landmarks, given the measured delays. This is achieved by an algorithm which combines gradient ascent and force-directed methods. We present experimental results to demonstrate the superior accuracy of the proposed geolocation scheme compared to previous methods.

The remainder of the paper is organized as follows. In Section II, we discuss the related work on geolocation schemes, particularly the Shortest Ping, GeoPing and CBG schemes. Section III develops a statistical approach to IP geolocation based on kernel density estimation and maximum likelihood estimation. Section IV presents the results of our experiments with datasets obtained from the PlanetLab [4] experimental network. Finally, section V concludes the paper with a summary of the main contributions of this work.

II. Related Work

In this section, we discuss and critique three earlier Internet geolocation schemes which are closest in spirit to our proposed statistical geolocation approach.

A. Shortest Ping

Shortest Ping (SPing) [1], [3] was one of the earliest attempts to use Internet measurements to geolocate a target host. In SPing, a set of hosts, called *landmarks*, perform network delay measurements by transmitting ICMP ping packets between each other. When a new target host is encountered, the landmarks determine their delays to the target. These delays are compared to the existing measurements.

More precisely, let $\mathcal{L}$ denote the index set for landmarks, i.e., the set of landmarks is given by $\{L_i : i \in \mathcal{L}\}$. The location of each landmark L_i, $i \in \mathcal{L}$, denoted by (φ_i, λ_i), is assumed to be known. Here, φ_i and λ_i represents the longitude and latitude, respectively, of landmark L_i in units of radians. Let $d_{i\tau}$ denote the delay from landmark i to the target τ. In Shortest Ping, the location estimate for the target is defined as (ϕ_k, λ_k), where

$$k = \arg\min_{i \in \mathcal{L}} \left\{ d_{i\tau} \right\}. \tag{1}$$

Since SPing depends only on the minimum RTT delay, an inaccurate measurement or a high speed link may have a significant impact on the estimated target location.

B. GeoPing

GeoPing [1] improves over SPing by introducing so-called *passive* landmarks in addition to the *active* landmarks used in Shortest Ping. Let $\mathcal{L}_a$ and $\mathcal{L}_p$ denote the index sets for *active* landmarks and *passive* landmarks, respectively. The index set for all landmarks is given by $\mathcal{L} = \mathcal{L}_a \cup \mathcal{L}_p$. The active landmarks indexed by $\mathcal{L}_a$ perform network delay measurements between each other and to the passive landmarks indexed by $\mathcal{L}_p$. Let d_{ij} denote the measured delay between landmarks L_i and L_j, where $i \in \mathcal{L}_a, j \in \mathcal{L}_a \cup \mathcal{L}_p$. In GeoPing, the location estimate for the target τ is defined as (ϕ_k, λ_k), where

$$k = \arg\min_{j \in \mathcal{L}} \left\{ \sum_{i \in \mathcal{L}_a} \left(d_{ij} - d_{i\tau}\right)^2 \right\}. \tag{2}$$

GeoPing is highly sensitive to outliers, since they are based on the minimum residual sum of squares. Thus, a large difference in the delay measurement to one target has a large impact on the Euclidean distance in delay space. In addition, both SPing and GeoPing estimate the position of the target in terms of the coordinates of one of the landmarks; therefore, when the target is relatively far from the "closest" landmark, the estimation error can be significant. Our proposed statistical geolocation algorithm is based on moving the estimated position in the direction of maximizing the likelihood function for the majority of the landmarks. Thus, measurements taken with respect to an individual landmark will have a much smaller impact on the final result than in SPing or GeoPing.

C. Constraint-Based Geolocation

Gueye *et al.* [2] proposed an approach to IP geolocation based on an idea from the field of sensor networks, called Constraint-based Geolocation (CBG). Their approach is based on the observation that the packet propagation speed over the Internet is at most the speed of light through optical fiber cable, which in turn is about 2/3 of the speed of light. This restriction induces circle-like bounds on the location of the target. If we denote the round trip time (RTT) between two hosts by d, an upper bound for the geographical distance between the two hosts is given by:

$$\hat{g} = \tilde{c} \cdot d, \tag{3}$$

where $\tilde{c} = 2/3 \cdot c$ and c is the speed of light (3×10^8 m/s). When the RTT is measured in ms and the geographical distance is measured in km, $\tilde{c}$ is approximately 100 km/ms; thus (3) can be written as

$$d = \frac{1}{100} \cdot \hat{g}. \tag{4}$$

This line is called the "baseline" in [2] and is illustrated in Fig. 3. All of the distance-delay measurements are situated above the baseline.

The distance upper bound provided by the baseline is too loose to be of any use. To tighten this upper bound, Gueye *et al.* [2] fit a so-called "bestline" $d = m_i \cdot g + b_i$, to each active landmark $i \in \mathcal{L}_a$. This bestline is the tightest bound below the distance-delay pairs. The constraints are: 1) all the distance-delay pairs should lie above the *bestline*, 2) the slope of the *bestline* should be at least as large as that of the *baseline*, and 3) the intercept of the *bestline* should be positive. Thus, determining the bestline can be formulated as a linear programming (LP) problem:

$$\begin{aligned} &\min_{m_i, b_i} \sum_{j \in \mathcal{L} \setminus \{i\}} d_{ij} - m_i g_{ij} - b_i, \quad \text{subject to:} \\ &d_{ij} - m_i g_{ij} - b_i \geq 0, \quad \forall j \neq i \\ &m_i \geq m \ \left(= \frac{1}{100}\right), \ b_i \geq 0, \quad i \in \mathcal{L}_a. \end{aligned} \tag{5}$$

For a given target τ, the CBG algorithm calculates the upper bounds based on the measured delays $d_{i\tau}$ from each active landmark L_i to the target (see Fig. 1). Thus, the *bestline* upper bound is

$$\hat{g}_{i\tau} = \frac{d_{i\tau} - b_i}{m_i}. \tag{6}$$

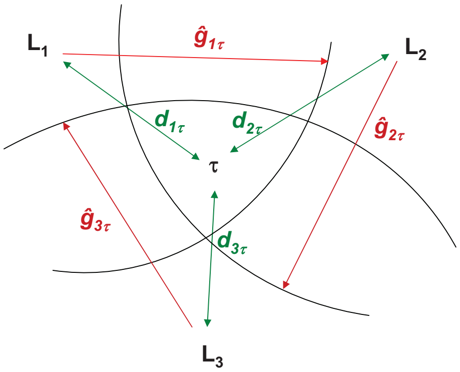

Fig. 1. Illustration of CBG scheme.

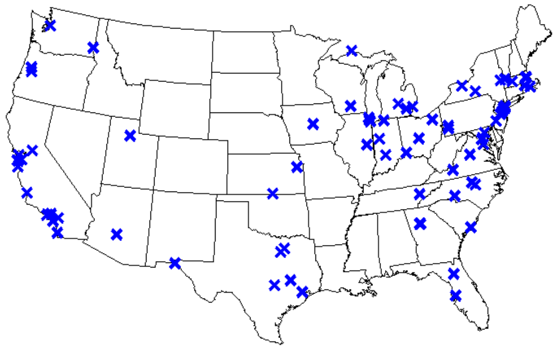

Fig. 2. Landmark distribution over the continental U.S.

The algorithm draws a circle with radius $\hat{g}_{i\tau}$ around each active landmark $i \in L_a$. By intersecting these circles, a region is obtained, where the target should be located. The location estimate of the target is taken to be the center of this region.

The area of the region is taken as a measure of confidence of geolocation. The authors claim their method is an improvement over the GeoPing method. However, the CBG method also has problems, such as large errors when the target is far from the landmarks, and it completely fails when even one of the distances is underestimated. The CBG scheme uses deterministic "upper" bounds imposed on the distances from the landmarks to the target to estimate the position of the target, but these upper bounds are not guaranteed to hold. Thus, in a significant number of cases, the region of confidence is either the empty set, or does not include the target. In contrast, our proposed algorithm uses the entire information given by measurements, by estimating the probability density function of the distance between landmarks and target, given the delay. This leads to improved accuracy by capturing the statistical variation of delays in the Internet.

III. Statistical IP Geolocation

In this section we develop a statistical approach to IP geolocation. Our approach consists of several steps. First, a "profile" of each landmark is constructed using the distance-delay pairs amongst the landmarks, resulting in a scatterplot for each landmark. Second, the joint probability distribution of the distance and delay is approximated using bivariate kernel density estimation. A Gaussian kernel is used for density estimation. Finally, a force-directed algorithm is used to obtain an estimate of the target location.

A. Construction of Landmark Profiles

The profile of an active landmark L_i, $i \in \mathcal{L}_a$ consists of the set of all distance-delay pair measurements originating at L_i towards the other (active or passive) landmarks L_j, where[1] $j \in \mathcal{L}\backslash\{i\}$. Our construction of landmark profiles is similar to that of Gueye *et al.* [2]. Multiple measurements are obtained between every pair of landmarks at different times, yielding the same distance, but different delays. In our experimental study, we used 85 servers in the PlanetLab research network [4]. The server locations are shown in Fig. 2. We obtained RTT measurements using the *ping* utility five times every 15 minutes for a period of one week, yielding up to $M = 282,240$ measurements for each target (in practice, not all measurements are successful). From the measurements, we obtained a scatterplot for each active landmark L_i, $i \in \mathcal{L}_a$ by taking delay measurements from L_i to all other landmarks (see Fig. 3).

For clarity of presentation, the scatterplot in Fig. 3 shows only the minimum delay measurements between *planet1.cs.stanford.edu* and 79 other PlanetLab nodes. The SPing and GeoPing methods use only minimum delay measurements. The CBG scheme uses the 2.5 percentile of measurements. By contrast, the statistical geolocation scheme proposed in this paper uses all of the delay measurement data for statistical analysis.

B. Kernel Density Estimation

Once the profile of each landmark is built, the second step is the estimation of the joint distribution of (G_i, D_i), where G_i represents the great circle distance between active landmark L_i, $i \in \mathcal{L}_a$ and the target τ, and D_i is the measured delay between L_i and τ. The joint probability density function of (G_i, D_i) is denoted by $f_{G_i,D_i}(g,d)$. The sample data to be collected is represented as follows:

$$\mathcal{S}_i = \left\{(g_{ij}, d_{ij}^{(l)}) : j \in \mathcal{L}_a, 1 \leq l \leq m\right\}, \quad i \in \mathcal{L}_a, \qquad (7)$$

where m is the number of delay measurements taken between a given pair of landmarks L_i and L_j. In our experiments, $m = 5 \times 4 \times 24 \times 7$, since 5 delay measurements from landmark L_i to landmark L_j, $j \neq i$, were taken once every 15 minutes over a period of one week. Let $M \triangleq |\mathcal{S}_i|$ denote the total number of delay measurements taken from a given landmark. For a set of 85 active landmarks, we have $M = 84m = 282,240$.

We apply the following kernel density estimator [5]–[7]:

$$\hat{f}_{i,\mathbf{H}}(g,d) = \frac{1}{M\det(\mathbf{H})} \sum_{j \in \mathcal{L}_a\backslash\{i\}} \sum_{l=1}^{m} \mathcal{K}((g - g_{ij}, d - d_{ij}^{(l)})\mathbf{H}^{-1}),$$

[1] Here, $A\backslash B \triangleq A \cap B^c$, where B^c denotes the complement of the set B.

where (g, d) is a vector consisting of great circle distance g and delay d; $\mathbf{H}$ is a nonsingular matrix, called the *bandwidth matrix*; and $\mathcal{K}$ denotes the kernel. In our experimental work, we use a diagonal bandwidth matrix and a Gaussian kernel:

$$\mathbf{H} = \left[\begin{array}{cc} h_1 & 0 \\ 0 & h_2 \end{array} \right], \; \mathcal{K}(g, d) = \frac{1}{2\pi} e^{-\frac{1}{2}(g^2+d^2)}. \tag{8}$$

Thus, the kernel density estimator becomes

$$\hat{f}_{i,\mathbf{H}}(g, d) = \frac{1}{2\pi h_1 h_2} \sum_{j \in \mathcal{L} \setminus \{i\}} \sum_{l=1}^{m} e^{-\frac{1}{2}\left[\left(\frac{g - g_{ij}}{h_1} \right)^2 + \left(\frac{d - d_{ij}^{(l)}}{h_2} \right)^2 \right]}. \tag{9}$$

Several methods are available for choosing the bandwidth parameters h_1 and h_2. Popular choices include various rules-of-thumb, bootstrap methods, plug-in methods, unbiased cross validation, and biased cross validation. Scott's rule-of-thumb is given by [5], [6]

$$\hat{h}_j = M^{-1/6} \hat{\sigma}_j, \quad j \in \{1, 2\}. \tag{10}$$

Although Scott's rule-of-thumb choice of bandwidth parameters makes normality assumptions of underlying unknown distribution, we prefer this method due to its low complexity, i.e., $O(M)$ as opposed to $O(M^2)$ for the other methods. This is especially important as we deal with large data sets (e.g., on the order of 250,000 samples).

C. Application of Force-Directed Method

We employ a force-directed algorithm as an approximation algorithm to maximize the likelihood of the target location estimate given the delay measurement data. The force-directed method iteratively applies a force on the target proportional to the gradient of the estimated conditional distribution of distance from each landmark to the target given the delay. At each step of the algorithm, the resultant of the forces from all landmarks is calculated and then the target location estimate is moved in accordance with the resultant force. Thus, our algorithm combines the force-directed method with gradient ascent optimization. The initial estimate of the target location can be set as the landmark with the shortest delay to the target.

The gradient ascent steps $\{\eta_i\}$ form a decreasing sequence converging to zero, to ensure the convergence of the force-directed method. The initial gradient ascent step η_0 is chosen to be such that the target is moved a given distance from its initial position (e.g., 100 km, which is the magnitude of 10^8 for the rule-of-thumb bandwidth). The algorithm stops when the target moved less than a value ϵ, where ϵ is chosen in such a way to achieve a tradeoff between computational overhead and accuracy requirement.

Since the landmarks and targets are located on the earth, great circle distances must be considered. We use the WGS-84 ellipsoid [8] as a model for Earth and apply the Vicenty formulas to compute great circle distances [9]. We have implemented the direct and inverse Vincenty's formula in two functions.

Direct Vincenty Formula:

$$((\varphi_2, \lambda_2), b_2) = v_{\mathrm{fwd}}((\varphi_1, \lambda_1), b_1, g), \tag{11}$$

which calculates the destination point (φ_2, λ_2) and the final bearing b_2 given the starting point (φ_1, λ_1), initial bearing b_1, and the great circle distance g from the starting point to the destination.

Inverse Vincenty Formula:

$$(g, b_1, b_2) = v_{\mathrm{inv}}((\varphi_1, \lambda_1), (\varphi_2, \lambda_2)), \tag{12}$$

which calculates the great circle distance g, the initial bearing b_1, and the final bearing b_2 given the starting point (φ_1, λ_1) and the destination point (φ_2, λ_2)

Our proposed force-directed steepest ascent algorithm is summarized as follows:

F1. Start with a guess of the latitude and longitude of the target $(\varphi_\tau^{(0)}, \lambda_\tau^{(0)})$. Initialize $k \leftarrow 0$.

F2. Calculate the distance and final bearing from each landmark to the target using the inverse Vincenty formula:

$$(g_i^{(k)}, b_i^{(k)}) \leftarrow v_{\mathrm{inv}}((\varphi_i, \lambda_i), (\varphi_\tau^{(k)}, \lambda_\tau^{(k)})), \quad i \in \mathcal{L}_a.$$

F3. Execute one step of gradient ascent:

$$l_i^{(k)} \leftarrow g_i^{(k)} + \eta_k \hat{f}'_{G_i|D_i}(g_i^{(k)} \mid d_{i\tau}), \quad i \in \mathcal{L}_a.$$

F4. For each $i \in \mathcal{L}_a$ calculate the force vector $\mathbf{F}_i^{(k)}$ as follows:

If $\hat{f}_{G_i|D_i}(l_i^{(k)} \mid d_{i\tau}) > \hat{f}_{G_i|D_i}(g_i^{(k)} \mid d_{i\tau})$ **then**

$$|\mathbf{F}_i^{(k)}| \leftarrow l_i^{(k)} - g_i^{(k)}; \; \mathrm{bear}(\mathbf{F}_i^{(k)}) \leftarrow b_i^{(k)}$$

Else $\mathbf{F}_i^{(k)} \leftarrow \mathbf{0}$.

F5. Calculate the resultant force vector

$$\mathbf{F} = |\mathcal{L}_a| \mathrm{gm}(\mathbf{F}_i^{(k)} : i \in \mathcal{L}_a)$$

F6. Move the target location estimate in the direction of the resultant force using the direct Vincenty formula:

$$(\varphi_\tau^{(k+1)}, \lambda_\tau^{(k+1)}) \leftarrow v_{\mathrm{fwd}}((\varphi_\tau^{(k)}, \lambda_\tau^{(k)}), \mathrm{bear}(\mathbf{F}), |\mathbf{F}|)$$

Increment k by one.

If target estimate moved more than ε **then** go to **F2**.

Else STOP

The conditional pdf estimate $\hat{f}_{G_i|D_i}(g|d)$ in **F3** and **F4** can easily be obtained from the joint kernel density estimate (9). In step **F4**, a force vector $\mathbf{F}$, in geographical coordinates, is represented as being comprised of a magnitude $|\mathbf{F}|$ and a bearing bear$(\mathbf{F})$. This is similar to the magnitude-phase representation in the complex plane. Referring to step **F5**, the operator gm$(\cdot)$ (geographical mean) computes the centroid of a set of points on a spherical surface. To compute the geographical mean, we make use of the built-in Matlab function MEANM. When the algorithm terminates in Step **F6**, the estimated location of the target is given by $(\varphi_\tau^{(k)}, \lambda_\tau^{(k)})$. The initial target location in Step **F1** can be obtained by applying a computationally simple geolocation method such as SPing or GeoPing [1], [3].

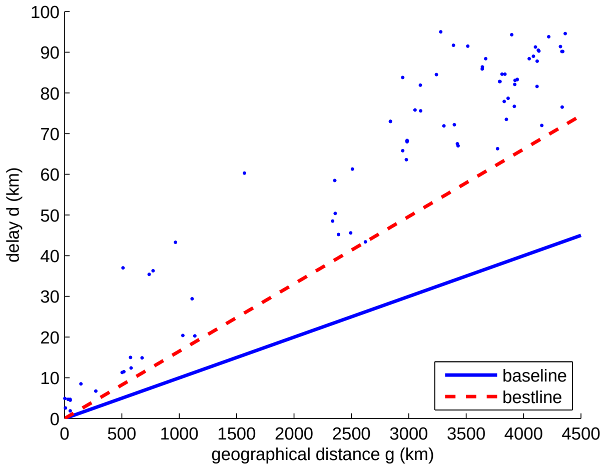

Fig. 3. Scatterplot of distance and delay from *planet1.cs.stanford.edu* to 79 other PlanetLab nodes across the U.S. (see Fig. 2).

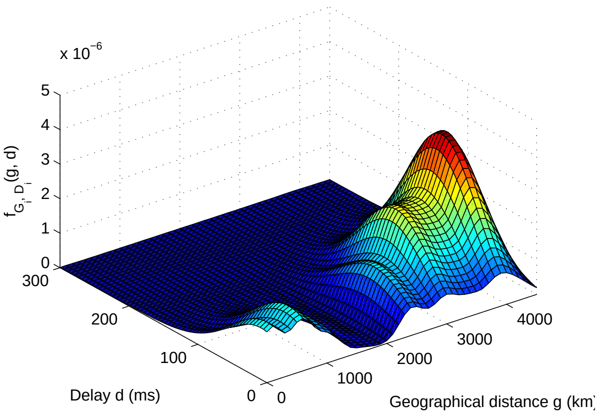

Fig. 4. Kernel density estimate of bivariate distribution of distance and delay using Gaussian kernel for *planet1.cs.stanford.edu*.

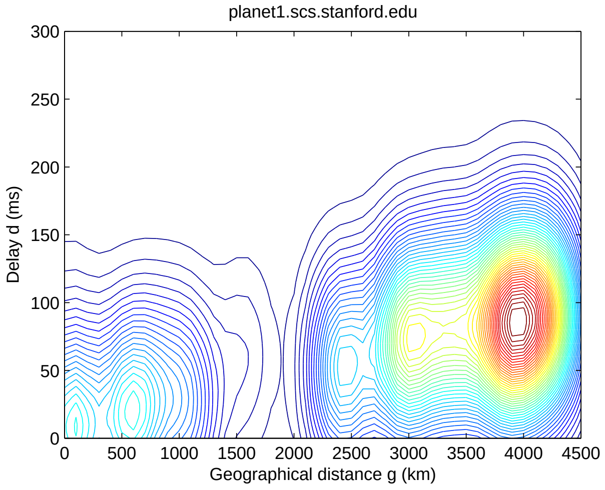

Fig. 5. Contour plot of kernel density estimate for *planet1.cs.stanford.edu*.

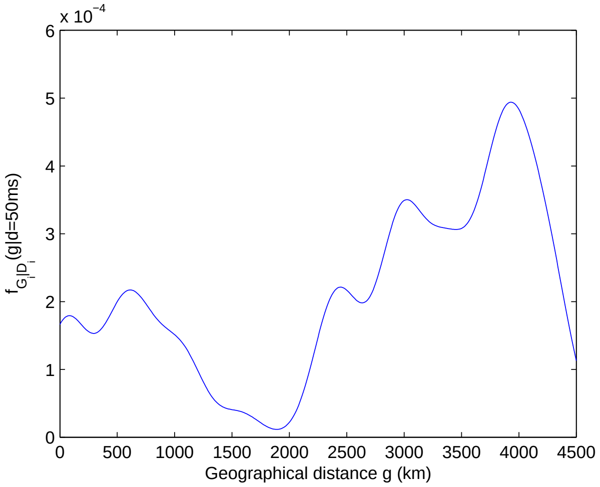

Fig. 6. Estimated conditional pdf of distance from *planet1.cs.stanford.edu* to a target, given a delay of 50 ms.

IV. Experimental Results

We conducted experiments over the PlanetLab network using 85 landmarks. The distribution of the landmarks over the continental U.S. is illustrated in Fig. 2. The PlanetLab database includes information on the latitude and longitude of each of the PlanetLab nodes. We used the CoMon project of PlanetLab to retrieve a list of the active nodes, filtered to select only one node per site. By means of a geocoding webpage written using JavaScript and the Google Map API, we filtered out a total of 93 sites located in the continental U.S. We tested each of these sites, of which only 85 nodes responded to *ping* commands (the others had firewall constraints).

We uploaded and executed a Python script in a distributed manner using the *codeploy* tool and saved the output in a log file. The log files were later downloaded and parsed using another Python script, and the measurement results were placed in comma separated value (CSV) files. As a result of our delay measurements over PlanetLab, we obtained 85 scatterplots and kernel density estimates of the joint pdf of distance and delay from each landmark to the target. Fig. 4 shows the KDE surface obtained at the PlanetLab node *planet1.cs.stanford.edu*. A contour plot of the kernel density estimate for the same landmark node is illustrated in Fig. 5. For this landmark, the conditional density of geographical distance given a 50 ms delay is shown in Fig. 6

The kernel density estimates were applied to the force-directed algorithm described in Section III-C to obtain the estimate of the target location. We validated the proposed geolocation scheme by removing each landmark from the set of all landmarks, and running our algorithms with the removed landmark as the target and the remaining landmarks. The initial target location estimate is the landmark which is closest from the point of view of RTT delay. The force-directed algorithm is designed to iteratively push the initial location estimate towards its true location, based on conditional distributions of geographical distance given delay. We observed that when

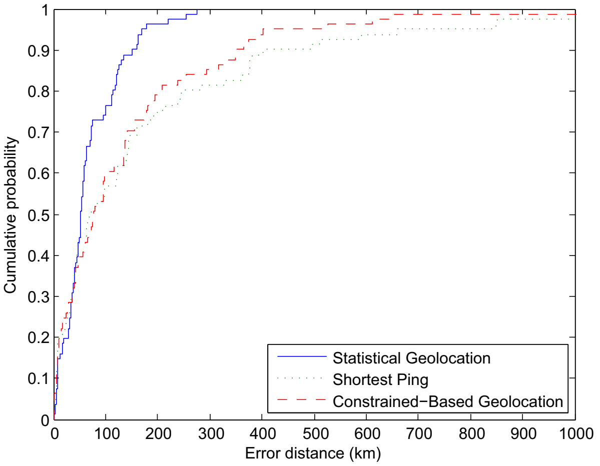

Fig. 7. Cumulative distribution function of estimation error: statistical geolocation (SG), CBG, and SPing.

the initial estimate is far from the real position of the target, our algorithm improves the estimate dramatically. However, when the initial estimate is close to the target, not much improvement is observed. To improve the resolution and accuracy, one has to increase the number of landmarks.

For comparison, we have implemented and executed the CBG and SPing algorithms. The CBG algorithm failed three times, yielding an empty confidence region. We removed these cases from the CBG statistics. In other five cases the confidence region did not include the target. By reducing the large errors, the average error of our statistical approach is 92 km. This is a dramatic improvement compared to 141 km for CBG and 184 km for SPing. The median error also decreases to 53 km, in comparison to 73 km for SPing and 78 km for CBG. We note that 77% of the location estimates from SG estimates were in the 100 km range, compared to 59% for CBG and 57% for SPing. Furthermore, 15% of the CBG estimates and 19% of the SPing estimations had an error of 300 km or more, while all but one of the SG estimates falled within the 300 km mark. Fig. 7 shows plots of the cumulative distribution function (cdf) of the estimation error for SPing, CBG, and SG. From this figure, it is clear that the statistical geolocation scheme is significantly more accurate than the CBG and SPing.

Table I displays the geolocation error performance of SPing, CBG, and SG. The error statistics shown in the left-hand column are the mean error, median error, maximum error, standard deviation of error, first quartile, and third quartile. All of the error values shown in the table are in units of km. In terms of mean error, SG shows a significant improvement over CBG, which in turn shows a significant improvement over SPing. The median errors of SPing and CBG are similar, while SG has a markedly smaller median error. Similarly, whereas the maximum error values for SPing and CBG are approximately the same, that of SG is about a factor of two smaller. Interestingly, the standard deviation of the error is smaller for CBG than for SG. The first quartile of the errors are approximately the same for all three schemes, but SG clearly outperforms the other two schemes in terms of the third quartile of error. In summary, the SG scheme appears to provide significantly higher accuracy than SG and CBG. There is however, room for improvement, as indicated by the result for the standard deviation of error. Aspects of the SG scheme that could be refined further include the kernel density estimation approach and the force-directed gradient ascent algorithm.

Error [km]	SPing	CBG	SG
mean	184	141	92
median	73	78	53
maximum	2167	2155	1054
std. dev.	309	176	238
1st quartile	30	28	32
3rd quartile	198	180	99

TABLE I
ACCURACY COMPARISON OF SPING, CBG, AND SG.

V. CONCLUSION

We proposed a statistical approach to geolocation of Internet hosts, based on a the collection of delay measurements among a set of landmark nodes. In contrast to earlier measurement-based geolocation schemes, which provide loose deterministic bounds on the target location, the proposed scheme captures the statistical variations in Internet delay measurements. Besides the collection of active delay measurements, the key elements of the approach include kernel density estimation to obtain an estimate of the joint density function of the geographical distances and delays between landmarks, and a force-directed algorithm to move the target location estimate towards a point that maximizes the likelihood function.

We conducted experiments over PlanetLab using 85 landmark nodes. Our results show a significant improvement in accuracy over the previous approaches, in particular, CBG and SPing.

REFERENCES

[1] W. N. Padmanabhan and L. Subramanian, “An investigation of geographic mapping techniques for Internet hosts,” in *ACM SIGCOMM*, 2001, pp. 173–185.

[2] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, “Constraint-based geolocation of internet hosts,” *IEEE/ACM Trans. Networking*, vol. 14, no. 6, pp. 1219–1232, 2006.

[3] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, “Towards IP geolocation using delay and topology measurements,” in *Internet Measurement Conference (IMC’06)*, 2006.

[4] PlanetLab, http://www.planet-lab.org.

[5] B. Silverman, *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1986.

[6] D. Scott, *Multivariate density estimation: theory, practice, and visualization*. Wiley-Interscience, 1992.

[7] W. Härdle, *Nonparametric and semiparametric models*. Springer, 2004.

[8] Geospatial Science Division, “World Geodetic System 1984,” U.S. Dept. of Defense, Tech. Rep. TR8350.2, July 1997, 3rd Ed.

[9] T. Vincenty, “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations,” *Survey Review*, vol. 22, no. 176, pp. 88–93, 1975.