GeoCET: Accurate IP Geolocation via Constraint-Based Elliptical Trajectories

Fei Du[1,2], Xiuguo Bao[3], Yongzheng Zhang[1,2(✉)], and Huanhuan Yang[3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{dufei,zhangyongzheng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] National Internet Emergency Center, CNCERT/CC, Beijing, China

Abstract. The geographical location of the IP device is crucial for many network security applications, such as location-aware authentication, fraud prevention, and security-sensitive forensics. Since most data mining-based methods are subject to the privacy protection policies, the delay-based measurement methods have broader application prospects. However, these methodologies are relying on heavyweight traffic on networks and high deployment costs. Besides, the worst case errors in estimation made by delay-based measurement methods render them ineffective. In this paper, we propose an accurate IP geolocation approach called GeoCET. This methodology only requires a small number of one-way delays (OWDs) to locate the targets, combining with elliptical trajectory constraints and maximum log-likelihood estimation technique. We introduce polynomial regression to fit the delay-distance model and enhance the accuracy of the localization. To evaluate GeoCET, we leverage real-world data which come from China, India, Western United States, and Central Europe. Experimental results demonstrate that GeoCET performs better for all existing measurement-based IP geolocation methodologies.

Keywords: Network security · IP geolocation · Delay-based measurement · Constraint-based elliptical trajectories

1 Introduction

Knowing the geographical location of Internet devices have an extensive range of applications, examples include delivery of local news and advertising, Internet anti-fraud (e.g., fraud signup, collision attack, brushing and spamming), credit card fraud detection, load balancing, resource allocation [9,10]. Particularly for law enforcement agencies, it is necessary to determine the location information very accurately as quickly as possible in order to satisfy all the requirements for an attacker's forensic strategy [24].

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 603–622, 2019.
https://doi.org/10.1007/978-3-030-30146-0_41

Our **goal** is to develop a high precision lightweight geolocation approach to locate active IP addresses on the Internet efficiently. Additionally, *IP geolocation* means determining the real-world location of an Internet-connected device. However, what makes this work challenging is that there is no one-to-one mapping between IP addresses and geographic locations. The dynamic assignment of IP addresses makes IP geolocation more difficult. On the other hand, because the propagation characteristics of the Internet are sharply influenced by factors such as the circuitous route, network congestion and queueing delay, it is more challenging to locate Internet devices. Besides, most Internet devices do not have the ability to self-positioning (e.g., Global Positioning System (GPS) or other location techniques [3]), other mobile devices may choose to hide location information due to privacy protection.

In the last years, the IP geolocation methods are based on static sources of information, such as registries and databases (e.g., [1,2]). However, with the adoption of IPv6, such databases become more difficult to update and maintain, the accuracy of these databases is in general not excellent. Some studies show that errors in the order of several thousand kilometers are possible [23].

For this reason, the device to be localized is mainly through network delay measurements (e.g., [8,27]). These methods for geolocation primarily measure end-to-end latency from a set of nodes with known locations of nodes to be geolocated using active probes (e.g., by using the Internet Control Message Protocol (ICMP), Ping or traceroute). Then, delays are converted into distances according to a previously defined delay-distance model, which assume that there is an existent correlation between network latency and geographical distance. (e.g., linear relationships include: *bestline*, $\frac{2}{3}c$ [22], $\frac{4}{9}c$ [16] or $\frac{3}{4}c$ [15]; non-linear relationships include probability distributions and hybrid strategies [5,6,25].) Finally, the coordinates of the target are inferred using geometrical techniques, such as [14,21,26] and [28]. Nevertheless, these methods do not achieve good accuracy, as inferences and approximations characterize the geolocation process, their accuracy is strongly dependent on the location of the landmark nodes to the target nodes. Therefore, these methods require a large number of available landmark nodes, relying on heavyweight traceroute-like or Ping-like probe packets on networks.

This paper proposes a novel accurate approach to IP geolocation—GeoCET. The GeoCET methodology considers two categories of nodes in the network: *Targets*, i.e., nodes with unknown geographic location that we aim to geolocate and respond to probes; *Analyszs*, i.e., nodes with known geographic location and the ability to send probe packets and receive response packets and perform localization operators. Then, we partition these nodes to "Observers" and "Landmarks". In prior literature, "Observers" sometimes referred to as "Vantage points", similar to Landmarks with probe capability.

The main contributions of this paper are summarized as follows.

– Presenting GeoCET, a novel approach for IP geolocation, which only relies on lightweight network load and reduces the number of feasible geographic coordinates to infer geographic location using elliptical trajectory constraints and maximum log-likelihood estimation technique.

– Constructing spoofed packets using the landmark's IP addresses to measure a target node, the one-way delay (OWD) links formed naturally can improve the localization accuracy. The flexibility and scalability of our scheme can effectively reduce the deployment of network resources.
– Evaluating GeoCET through detailed experiments on the real-world network, with nodes based in China, India, Western United States, and Central Europe. The evaluation involves analyzing the algorithm's accuracy and processing time, and other factors that affect the efficacy of GeoCET's results.

The rest of the paper is organized as follows. Section 2 presents the most relevant work in the field. Section 3 describes the delay-distance model and the detail of GeoCET algorithm. An empirical evaluation of GeoCET and a security discussion are presented in Sects. 4 and 5 respectively. Finally, Sect. 6 concludes the paper, and future work is outlined.

2 Related Work

In this section, we review related work which is closer in spirit to our proposed geolocation approach.

The constraint-based-geolocation (CBG) [14] used the limit of "*bestline*" to compensate for the detour and bloat of routes on the Internet. However, since it is difficult to predict whether a route from a monitor node to a target node is detoured, CBG is usually useful only when the target node is close to the monitor node. Another geolocation system that used information about intermediate routers is Topology-based Geolocation (TBG) [16].

Li et al. [19] developed a simple IP address mapping scheme GeoGet, a large number of web servers are used as passive landmarks, and the target maps to the geographic location of the landmark with the shortest delay. In order to control the measurement overhead, a multi-step detection method is used to optimize the geographical location of the target. The Octant [26] framework used a variety of information to locate the target node. It divides all information into positive and negative constraints to narrow the prediction area and improve localization accuracy. A positive constraint refers to an area where the target node may be located, and a negative constraint refers to a node cannot be located. It does not locate the position of the target node at a specific coordinate but represents its possible position as a surface determined by a Bézier curve. Since the network delay does not conform to the ideal delay transmission model, Octant introduces a "height" dimension to represent the access delay of the last hop.

The Spotter [18] algorithm is based on a detailed statistical analysis of the relationship between network delay and geographic distance and uses a probabilistic method to derive a general delay distance model, which can be achieved by the parameter estimation. In the case of multiple nodes, the delay distribution represents the joint probability of the independent normal distribution, and the parameter estimation method of the normal distribution is relatively simple. Compared to the same kind of active measurement method (relying on a large amount of network load), Posit [11] only requires a small amount of Ping

measurement, combined with computational useful statistical embedding technology to locate the target node. Consequently, the computational complexity of the Spotter and Posit is small. Hillmann et al. [15] presented a new approach for optimizing the Landmark position for active measurements—Dragoon. For a reasonable Landmark selection is crucial for highly accurate localization services, the goal is to find landmarks close to the target in terms of infrastructure and hop count. Besides, they introduced an improved approach to adaptability and more accurate modeling of the geolocation process. Whereas, the number of samples and representativeness are important factors affecting the accuracy of estimation.

CPV [4] has been proposed as a delay-based mechanism that verifies clients' geographic locations, and they introduced a new OWD-estimation algorithm and evaluated its practicability by the probability distribution of one's absolute error. Compared with the round-trip halving, the one-way delay is more accurate in many scenarios.

GeoCET differs from the approaches as mentioned above, because it uses a lightweight network load to achieve higher accuracy and is easy to deploy and implement. Computational complexity also has significant advantages over similar measurement-based IP geolocation methodologies. We will describe in detail in the next section.

3 GeoCET Geolocation Methodology

This section presents some definitions about this paper, application scenario, and the detail of the GeoCET algorithm.

3.1 Notations and Definitions

We present the relevant concepts and the formalized description of the problem.

Measurement Delay (x): refers to the round-trip delay directly measured between nodes or the time interval between the request package and response package, which mainly refers to the propagation delay [17] between nodes, ignoring the transmission delay and processing delay. e.g., the interval between the request SYN packet and the response ACK packet of the TCP protocol, the packet sending of UDP protocol on higher port and the delay of the response packet.

In this paper, we use **one-way delay** (OWD) as the measurement delay. Let R be the number of value collected, and let $x_i = \{x_{i,1}, x_{i,2}, ..., x_{i,R}\}$ be the set of one-way delay between node i and the target.

Steady-State Delay ($\hat{x}$): we assume that there is a steady state of delay between network nodes, i.e., when the expansion of delay caused by network load and processing time of intermediate nodes is excluded, the propagation delay between any two nodes is a specific value. Given that the network is a dynamic system, the delay between nodes is also dynamic, so the minimum

value in multiple measurements is chosen to be the steady-state delay between the two nodes. Let us define $\hat{x}_i = \min(x_i)$, then $\hat{x}_i$ is used for computing the steady-state delay between node i and the target.

Problem. In this paper, we ask the following question: is it possible to design an algorithm to achieve a high-precision IP geolocation algorithm with a fine-grained city block scale? In addition to being accurate and fast response time, such algorithms should also be scalable to networks of different application scenarios, and flexible in its use of computing resources.

For theoretical analysis purposes, we consider the following **scenario**. There are analysis nodes set $\mathcal{A}$ to measure target nodes set $\mathcal{T}$ ($\mathcal{T} = \{t_1, t_2, ..., t_l\}$), $\mathcal{A} = \mathcal{V} \cup \mathcal{L}$. where $\mathcal{V}$ ($\mathcal{V} = \{i_1, i_2, ..., i_m\}$) is a subset of analysis nodes as *observers* to send probe packets, and $\mathcal{L}$ ($\mathcal{L} = \{j_1, j_2, ..., j_n\}$) is a subset of analysis nodes as landmarks receive response packets of target nodes. Analysis nodes set $\mathcal{L}$ are used as *landmarks* and their distribution is over a small scale area (e.g., within 50 km), it satisfies the normal distribution. The network topology connectivity of analysis nodes set $\mathcal{A}$ can be approximated as the cyberspace with a 2-dimensional Euclidean model.

Ciavarrini et al. [7] derive the Cramér-Rao low bound (CRLB) of IP geolocation with delay-distance model. They proved that the distance between the landmarks related to the geolocation and the target should not be too large. Consequently, we limit the geolocation scenario to a range of 50 km.

The GeoCET algorithm will estimate the geographic location of each target node using only these steady-state delay measurement vectors from a set of analysis nodes.

3.2 Delay-Distance Model

According to the steady-state delay between the analysis nodes, combined with the geographical location of the known nodes, it can be drawn that the conversion relationship between the distances and delays of the analysis nodes. We define $\boldsymbol{r} = [r_1, r_2, ..., r_N]^T$ the vector of measured distances between target and the analysis nodes, and $\boldsymbol{d} = [d_1, d_2, ..., d_N]^T$ the vector of real distances between target and the analysis nodes. Ranging information can be modeled as in Eq. (1):

$$\boldsymbol{r} = \boldsymbol{d} + \boldsymbol{\delta} \tag{1}$$

with $\boldsymbol{\delta} = [\delta_1, \delta_2, ..., \delta_N]^T$ is the vector of errors associated to the ranging phase, when $\boldsymbol{\delta}$ is not a zero vector.

There is often a certain degree of error in the end-to-end direct measurement delay or the estimated relative delay. The conversion function calculated based on the steady-state delay between the analysis node and the neighbor nodes (which can be regarded as neighbor nodes between the analysis nodes) can eliminate some errors, so we use the steady-state delay to locate the target IP address.

When the observer is far away from the target IP, the delay is easily affected by factors such as network load and routing policies, and the delay measurement

value is prone to be too large, which makes it difficult to obtain accurate distance constraints based on the *bestline* method. It makes the distance of the cyberspace violate the triangular inequality of the Euclidean space in the measurable region.

Additionally, the factors affecting delay accuracy also include node jitter, coordinate drift, non-shortest route, malicious attacks, and link delays and so on.

Therefore, we limit the distance between the *observer*, the target IP and the analysis nodes on a relatively small scale to minimize the impact of the delay error. In theory, the principle is satisfied: (1) in the measurable region, the distance in the cyberspace must conform to the triangular inequality of the European space; (2) the propagation delay occupies a large proportion in the steady-state delay, so that the conversion relationship between the steady-state delay and the distance can be obtained by the least squares method.

Considering the non-linear relationship of the delay distance, we use a polynomial regression model [20] to solve the delay-distance conversion relationship. Suppose the conversion relationship is polynomial (2):

$$f_\rho(\hat{x}) = \rho_1 \hat{x}^n + \rho_2 \hat{x}^{n-1} + \rho_i \hat{x}^{n-i+1} + ... + \rho_{n+1} + \epsilon \tag{2}$$

where n is the degree of polynomials, $\hat{x}$ is the steady-state delay, ρ_i is the conversion coefficient, and $f_\rho()$ is the distance calculated from the delay. Since the regression function is linear in terms of the unknown coefficients $\rho_1, \rho_2, ...$ Therefore, for least squares analysis, the computational and inferential problems of polynomial regression can be completely addressed using the techniques of multiple regression.

Conveniently, the polynomial regression model (2) can be expressed in matrix form in terms of a design matrix $\mathbf{X}$, a distance vector f_ρ, a coefficient vector $\vec{\rho}$, and a vector $\vec{\epsilon}$ of random errors. Which when using matrix notation is written as:

$$\vec{f_\rho} = \mathbf{X}\vec{\rho} + \vec{\epsilon} \tag{3}$$

The vector of estimated polynomial regression coefficients is

$$\widehat{\vec{\rho}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\vec{f_\rho} \tag{4}$$

Since $\mathbf{X}$ is a Vandermonde matrix, the invertibility condition is guaranteed to hold if all the $\hat{x_i}$ values are distinct. This is ordinary least squares estimation (OLSE) solution [13].

3.3 Position Estimation Using Elliptical Trajectory Constraints

In contrast to delay-based and statistical-based methods, our algorithm's expectation is to locate the target IP address in a relatively small region (e.g., block in the city). Unlike machine learning-based methods (e.g., [12,26]), there is no need to explicitly define population and geographic data as input to the algorithm. We only consider the known locations of analysis nodes (as viewed landmarks and observers) in the infrastructure contained in the geographically constrained region C, where we expect to exploit regional information with high Internet resource density.

Therefore, given the set of possible coordinates (lat, lng) in the region C found by constraint-based geolocation, which embeds into the cyberspace with a 2-dimensional Euclidean model by the elliptical trajectory.

We define the set of Internet resource nodes (described in Sect. 4), aim to use the observer node i, the landmark nodes set $\mathcal{L}$ to geolocate the target node t. The main process of the GeoCET algorithm is as follows.

(1) Clock synchronization is performed on the analysis nodes involved in the location target node t.
(2) We perform end-to-end mutual measurement on nodes in node i and set $\mathcal{L}$, measure the steady-state delay in the current network situation, and calculate the delay-distance conversion relationship $\vec{f_\rho}$ (in Eq. (3)) and polynomial regression coefficient vector $\widehat{\vec{\rho}}$ (in Eq. (4)).
(3) The node i spoofs one's IP address with node j ($j = 1, 2, ..., n$) to send probe packets to the target node t, the target node t responds to the response packet to the corresponding IP node j, respectively. The link sequence of the packet is: node $i \rightarrow$ node $t \rightarrow$ node j, computing its one-way steady state delay $\hat{x}_{i,t,j}$, i.e., $\hat{x}_{i,t} + \hat{x}_{t,j}$.

We combine the delays $\hat{x}_{i,j}$ of nodes i and j to analyze whether $\hat{x}_{i,t,j}$ is greater than $\hat{x}_{i,j}$, if $\hat{x}_{i,t} + \hat{x}_{t,j} \not> \hat{x}_{i,j}$, i.e., the delay distance violates the triangular inequality of Euclidean space, ignoring the measured value of the link.
(4) In contrast to previous work, we do not adopt the intersection area of N circles as the candidate area of the target node, but exploit the elliptical trajectory intersection area, which takes the nodes i and j as the focus, and the distance sum after the delay conversion is constant, as shown in Equation (5) and Fig. 1.

$$\zeta_{i,t} = \bigcap_{j=1}^{n} E_{i,j} \left\{ \left(F_i, F_j\right), |F_iT| + |TF_j| = 2a, (2a > |F_iF_j|) \right\} \tag{5}$$

where a is a constant, $E_{i,j}$ is an ellipse with F_i and F_j as the focus, the trajectory of the moving point $T(x, y)$ is the possible position of the target, $\zeta_{i,t}$ is the intersection of multiple elliptical trajectories, The center of the region $\zeta_{i,t}$ is taken as the target node geolocation, denoted as $\ell_{i,t}$. Other variables are as follows.

$$\begin{aligned}
&F_i = (-c, 0), F_j = (c, 0), T = (x, y) \\
&|F_iF_j| = 2c = f_\rho\left(\hat{x}_{i,j}\right) \\
&|F_iT| = \sqrt{(x+c)^2 + y^2} = f_\rho\left(\hat{x}_{i,t}\right) \\
&|TF_j| = \sqrt{(x-c)^2 + y^2} = f_\rho\left(\hat{x}_{t,j}\right) \\
&2a = \sqrt{(x+c)^2 + y^2} + \sqrt{(x-c)^2 + y^2} = f_\rho\left(\hat{x}_{i,t,j}\right)
\end{aligned}$$

The intersection points of the elliptical trajectories that have been modeled between the target node and analysis nodes using the probe packet

paths is shown in Fig. 2, in the case (c), due to the symmetry caused by the collinearity of the analysis nodes in the geometric space, the localization of the target node is determined to be false positive. When elliptical trajectories intersect straight lines, there is a false positive position in case (d) but not in the case (b). Therefore, it is necessary to avoid the phenomenon of multi-node collinearity as much as possible.

(5) By traversing multiple observer nodes i in set V, we will get a set of candidate regions Ω_t of the target node t, $\Omega_t = \{\ell_{1,t}, \ell_{2,t}, ..., \ell_{d,t}\}$, $|\Omega_t| = d$, then, use the statistical algorithm to find the location of the target node t, $\hat{\ell}_t$, by maximizing the log-likelihood given measurements $\ell_{k,t}$ from the analysis nodes and to target node.

$$\hat{\ell}_t = \arg\max_{\ell_t \in \Omega_t} \hbar_i \left(\ell_t\right) = \arg\max_{\ell_t \in \Omega_t} \sum_{i=1}^{d} \log P\left(\ell_{i,t} | \hat{\omega}_i\right) \tag{6}$$

where $P\left(\ell_{i,t}|\hat{\omega}_i\right)$ is the posterior probability,

$$P\left(\ell_{i,t}|\hat{\omega}_i\right) = \frac{P\left(\hat{\omega}_i|\ell_{i,t}\right) P\left(\ell_{i,t}\right)}{\sum P\left(\hat{\omega}_i|\ell_{i,j}\right) P\left(\ell_{i,j}\right)} \tag{7}$$

The $\hat{\omega}_i$ is the coordinate (lat_i, lng_i) of the observer node i, then $P\left(\ell_{i,t}\right)$ indicates the prior possibility of a candidate region. The set of of analysis nodes that lie in the constrain region, C.

An example of the scenario using GeoCET methodology can be found in Fig. 1. Considering that only four analysis nodes are needed to build an intersection region using the one-way delay, the computational complexity of our proposed algorithm is very low, and the traffic generated by the measurement is negligible.

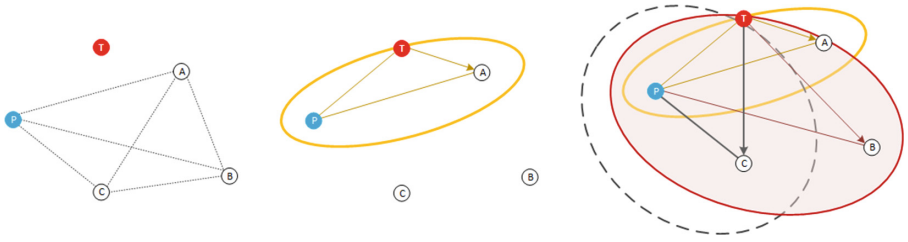


Fig. 1. (Left) - Example: probe node is P, analysis nodes are A, B, C, and the target node is T. (Center) - The P spoofing A's IP address measures T, $delay_{PT} + delay_{TA} = 2\beta\left(delay_{PT} + delay_{TA} > delay_{PA}\right)$, β is a constant, and the candidate trajectory of T is an ellipse with P and A as the focus. (Right) - P spoofs one's IP address through A, B and C to measure the target T separately. The intersection of the three elliptical trajectories is the position area of T.

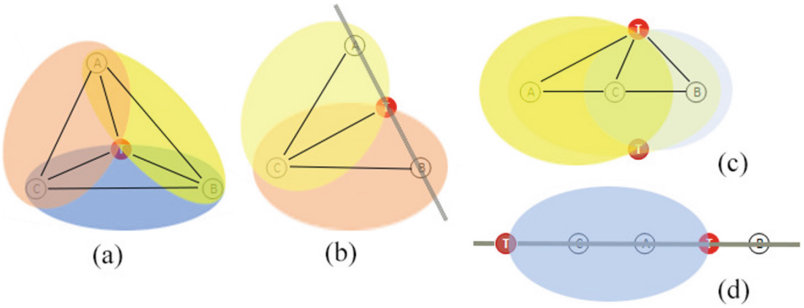


Fig. 2. Different cases of intersection: (a) One intersection point of multiple elliptical trajectories. (b) One intersection point of multiple elliptical trajectories and a straight line. (c) Two intersection points of multiple elliptical trajectories. (d) Two intersection points of elliptical trajectory and multiple lines.

3.4 GeoCET Geolocation Algorithm Summary

The complete GeoCET geolocation methodology is presented in Algorithm 1. The nodes in the set $\mathcal{A}$ ($\mathcal{A} = \mathcal{V} \cup \mathcal{L}$) are required to be an approximately uniform distribution with known geographic location, the networks are connected to each other, and the space constructed by the steady-state delay $\hat{x}$ conforms to the 2-dimensional Euclidean space.

To prevent overfitting and multiple feasible solutions, we carefully select the analysis nodes that participate in building the polynomial regression model. In practice, we construct a steady-state delay matrix $\mathbf{M}$ through multiple measurements in batches, minimizing the effects of cumulative errors.

Usually, $m < n$, we need fewer observers than landmarks. In the best case, where m is 1, and n is merely 3. This GeoCET algorithm achieves high precision with fewer probe packets, and the traffic on these networks is negligible.

In order to make the algorithm have a satisfactory convergence speed, the errors or outliers should be eliminated from the set $\mathcal{V}$ and $\mathcal{L}$.

GeoCET comprises two high-level capabilities (in Fig. 3): *Delay-distance generation* takes a geographic region that a target IP address belongs to as input, and generates a parameter vector of polynomial regression (PR) model for localization. *Candidate landmark localization* takes the target IP address and PR model as input and generates the target IP positions in the specified geographic region as output.

4 Evaluation

To validate and evaluate our GeoCET geolocation methodology, we exploit four datasets. In this section, we compare GeoCET with prior work and analyze relevant experimental results.

Algorithm 1. GeoCET Geolocation Algorithm

Input:

The set of *Observers*, $\mathcal{V} = \{i_1, i_2, ..., i_m\}$, $|\mathcal{V}| = m$.
The set of *Landmarks*, $\mathcal{L} = \{j_1, j_2, ..., j_n\}$, $|\mathcal{L}| = n$.
The set of *Targets*, $\mathcal{T} = \{t_1, t_2, ..., t_l\}$, $|\mathcal{T}| = l$.

Output:

The geographical location of each target IP in the set $\mathcal{T}$.

1: **for** each $i \in [1, m]$ and $j \in [1, n]$ **do**
2: initialize a steady-state delay $\hat{x}_{i,j}$ between node i and node j;
3: build a steady-state delay matrix $\mathbf{M} = (\hat{x}_{i,j})_{m \times n}$;
4: determine the delay-distance conversion polynomial regression function $\vec{f}_\rho$ and the conversion coefficient vector $\widehat{\vec{\rho}}$ using Equation (3) and (4);
5: **end for**
6: **for** each $t \in [1, l]$ **do**
7: use Equation (5) to resolve the each t intersection regions Ω_t;
8: **while** $(|\Omega_t| > 1)$ **do**
9: select the maximum log-likelihood estimation $\hat{\ell}_t$ from Ω_t using Equation (6) and (7);
10: **end while**
11: **if** $(\Omega_t \neq \phi)$ and $(|\Omega_t| = 1)$ **then**
12: the element ℓ_t in Ω_t is the geographic location of node t;
13: **end if**
14: **end for**

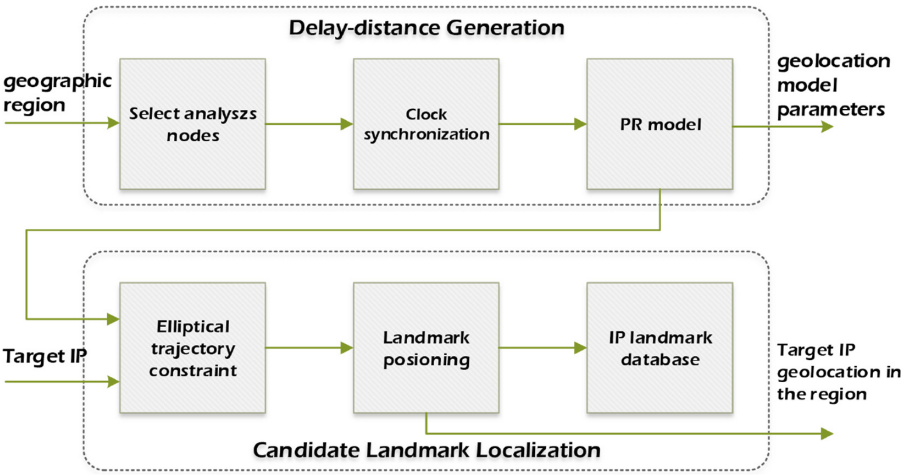
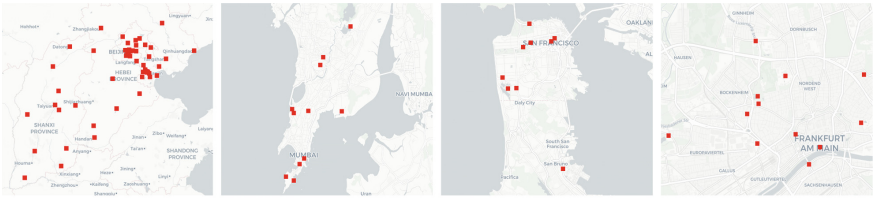


Fig. 3. GeoCET Components.

4.1 Experimental Data

We use a set of measurements collected from 361 analysis nodes with ground truth location knowledge. These nodes include: 75 cloud server nodes for Alibaba Cloud lease and their accurate locations are verified using codes written in Python and accessed the Google Map API, and volunteers provide the accurate location of 286 servers. Figure 4 shows the geographical distribution of the analysis nodes.



(a) (b) (c) (d)

Fig. 4. Analysis nodes come from four different regions: (a) analysis nodes in China; (b) analysis nodes in Mumbai; (c) analysis nodes in Silicon Valley and (d) analysis nodes in Frankfurt.

In the area where the analysis nodes are being, we collected more than 40,000 active IP addresses based on the principle of crowdsourcing. These IP addresses have fine-grained location information, i.e., latitude and longitude coordinates in the WGS84 coordinate system. Table 1 shows the specific information of the target nodes.

Table 1. Target nodes come from different regions of the world.

Different regions	Target nodes (crowdsourcing)
Beijing, Tianjin, Hebei, Shanxi, China	32,719
Mumbai, India	1,768
Silicon Valley, Western United States	2,574
Frankfurt, Central Europe	3,693

Among these target nodes, IP addresses in China cover different scenarios where the analysis nodes are dense or sparse; IP addresses in Mumbai are relatively complex in network topology; IP addresses in Silicon Valley belong to high-speed network connectivity; moreover, the geographical distribution of the analysis nodes and target nodes in Frankfurt is approximately uniform.

4.2 Implementation

We implemented GeoCET algorithm in C and Python. The probing packets are marked with the characteristic words as the fingerprint in the content field.

Each packet contains the IP address of the target node (reflection device) and the original packet transmission time, which is on the path from the observer to landmark nodes. Our implementation has six components (Fig. 3). All experiments described in this paper are run on analysis servers with Intel Xeon (Skylake) Platinum 8163 at 2.5 GHz, 16 GB memory and 40 GB hard drives. Below, we discussed the primary components of parallelizing GeoCET computing across multiple servers.

(1) PR model. This component uses a vector of one-way delays (OWDs) to solve the coefficient vector of the polynomial regression model, as polynomials have broader representation capabilities in a global network. Compared to the round-trip delays, One-way delays (OWDs) can mitigate the effect of network instability, path asymmetry and so on. It uses TCP-based measurements instead of Ping-based. The fundamental reasons are (1) routers that block ICMP, and (2) firewall constraints the packets.
(2) Elliptical trajectory constraint. It uses the method of spoofing the peer IP address and requires distributed collaboration for measurement. The geometry of the elliptical trajectory is the optimal choice for this mechanism.
(3) IP landmark database. The database maintains an IP address corresponding to the physical location information of the network entity, in addition to the latitude and longitude coordinates of the GPS, and also includes a semantic description of the geographic location. The network entity landmark database is dynamic.

4.3 Metrics

To measure the performance of GeoCET, we use two metrics: *accuracy* and *processing time*. We discuss the false positive rate of GeoCET, which can be used to determine GeoCET's precision.

The *accuracy* of GeoCET is measured by its positioning error, the distance between GeoCET's position $g(x_i)$ and ground truth $\boldsymbol{d}$. The value of root mean square error (RMSE) is used to calculate the metric as follows:

$$RMSE(\boldsymbol{x}, g) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(g(x_i) - \boldsymbol{d})^2}$$

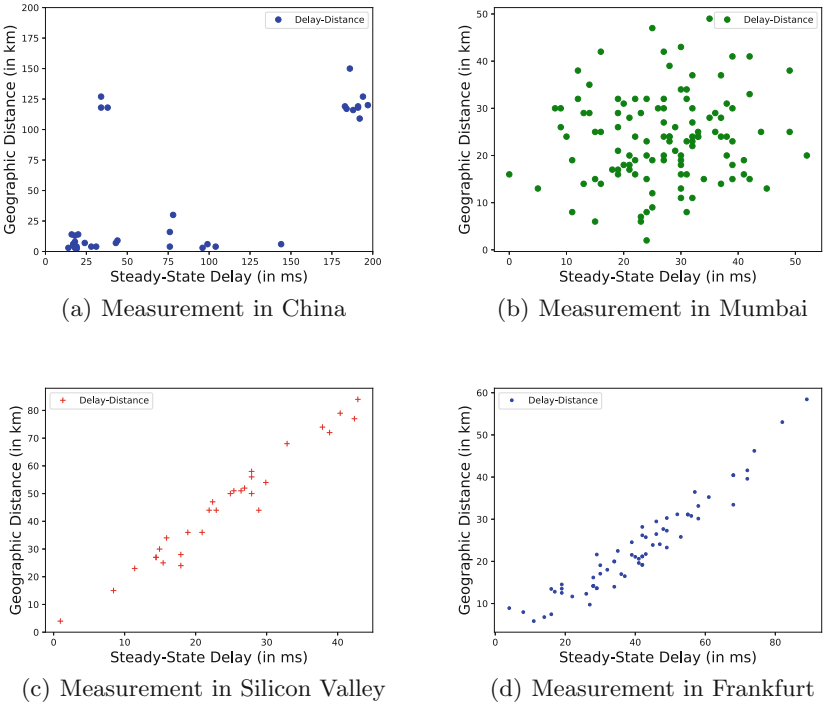
For *processing time*, we quantify the processing speed of each component in GeoCET, which can depend on various factors such as the adopted communication technologies. Since the processing time is a relative value, we define the measured average of a set of data in the same network environment as the evaluation metric.

$$PT = \boldsymbol{\lambda}\frac{1}{m}\sum_{i=1}^{m}\left(t^i_{DDG} + t^i_{CCL}\right)$$

where t^i_{DDG} is the measurement time of *Delay-distance Generation* component, t^i_{CCL} is the computation time time of *Candidate Landmark Localization* component, $\boldsymbol{\lambda}$ is the adjustment factor of the network environment.

4.4 An Example: Delay-Distance Model

In the delay-based measurement method, a large number of errors may result due to the non-linear relationship between the network distance and the geographical distance. For a more explicit expression, the scatter plot of the delay-distance is shown in Fig. 5, which only use the steady-state delay between the analysis nodes in datasets.



(a) Measurement in China (b) Measurement in Mumbai

(c) Measurement in Silicon Valley (d) Measurement in Frankfurt

Fig. 5. Example: Scatter plot of delay and distance from four datasets.

We solve polynomial regression function $\vec{f_\rho}$ in Python environment, the specific index is shown in Table 2. In general, the closer the R^2-*score* coefficient is to 1, the smaller the value of root mean square error (RMSE), the better the fitting effect of the corresponding polynomial. As can be seen from Table 2, we choose a polynomial fit of $degree = 5$. The conversion relationship is: $f_\rho(\hat{x}) = 9.67954503e{+}00\hat{x}^5 - 1.78578444e{-}01\hat{x}^4 + 1.22928376e{-}03\hat{x}^3 - 3.27890994e{-}06\hat{x}^2 + 2.74087855e{-}09\hat{x} - 112.21036574$.

In addition, in Fig. 5, the delay-distance relationship corresponding to the measurement dataset in Mumbai conforms to the normal distribution, and a linear function can represent the delay-distance relationship corresponding to the measurement dataset in Silicon Valley, and the measurement dataset in Frankfurt corresponds to the delay-distance relationship can be described by a second-order polynomial function, i.e., $degree = 2$.

Table 2. Polynomial conversion relationship from measurement dataset in China.

degree	*RMSE*	R^2-*score*
1	53.80	0.12
2	40.22	0.51
3	39.98	0.51
4	37.73	0.57
5	30.83	0.71
6	33.51	0.66
7	35.50	0.62
8	35.21	0.62

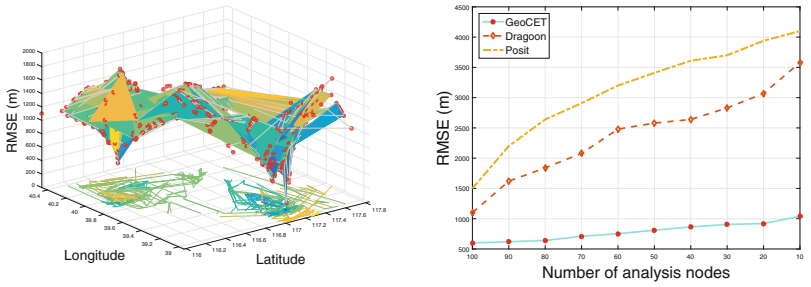
In some scenarios, polynomials do not well describe the relationship between steady-state delay and geographic distance. The main reasons are: (1) In the process of creating a steady-state delay matrix **M**, it accumulates more error. (2) Polynomial regression does not always describe the relationship between network delay and real geographic location. (3) Polynomial coefficient vector $\widehat{\vec{\rho}}$ has no solution or multiple feasible solutions.

4.5 Accuracy

The accuracy of the geolocation algorithm is central to GeoCET. Figure 6a shows the value of RMSE in a scenario with ten analysis nodes and a set of target nodes in China, this figure depicts the GeoCET's accuracy (approximately 500–1000 m). "Peak" and "Valley" are caused by the non-uniform distribution of analysis nodes and the dynamics of the network in China.

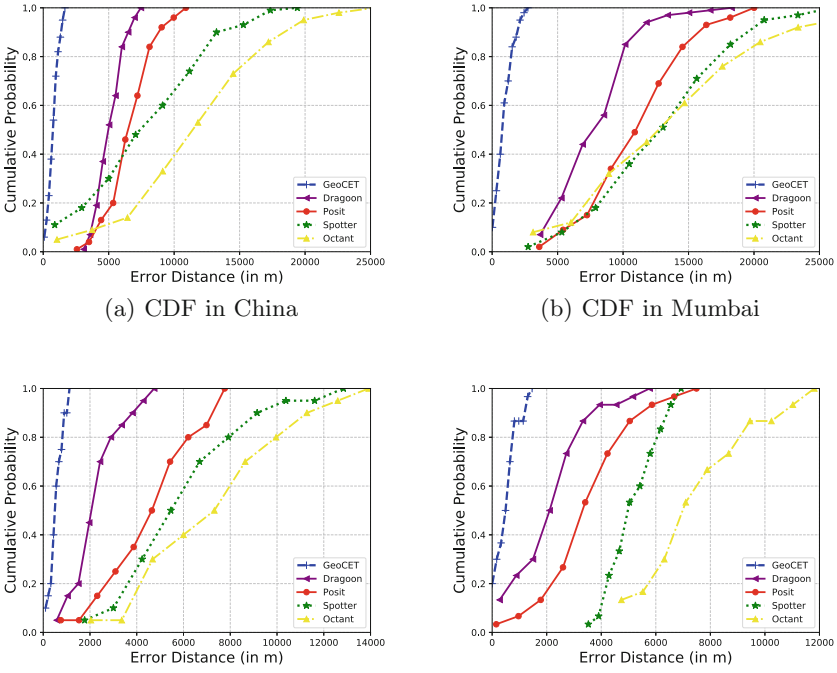
To further evaluate the accuracy of the GeoCET algorithm, we consider the probability distribution of analysis nodes (Landmarks and Observers) in different scenarios. In four different regions, we randomly divide the analysis nodes and target nodes into five groups. In the same measurement environment, we compared the GeoCET algorithm to Octant [26], Spotter [18], Posit [11] and Dragoon [15]. We did not get the source codes of these related algorithms. For the comparison of experiments, we implemented the core functions of the related algorithms based on the description in the literature. Taking the mean value of multiple measurements of five sets of data as input, Fig. 7 shows the cumulative probability distribution function (CDF) of errors of correlation comparison algorithms.

We find that the GeoCET outperforms the other algorithms in these four datasets. Regarding Dragoon, it is sensitive to the performance of the network, and the performance difference is significant in different networks. The Posit algorithm has a higher accuracy than Spotter and Octant, but still less accuracy than the GeoCET algorithm. Table 3 shows the comparison results of the median errors for these algorithms.



(a) (b)

Fig. 6. (a) Accuracy: 3D representation in China and (b) Accuracy when varying the number of analysis nodes.



(a) CDF in China (b) CDF in Mumbai

(c) CDF in Silicon Valley (d) CDF in Frankfurt

Fig. 7. Cumulative probability distribution of localization error for different methodologies in four datasets.

Table 3. The results of five algorithms in different measurement datasets.

Different datasets	GeoCET	Dragoon	Posit	Spotter	Octant
Data in China	870 m	5,436 m	7,630 m	10,136 m	14,423 m
Data in Mumbai	1,029 m	9,130 m	13,429 m	14,011 m	17,510 m
Data in Silicon Valley	580 m	3,128 m	4,685 m	7,331 m	8,347 m
Data in Frankfurt	620 m	2,536 m	4,422 m	5,536 m	7,949 m

4.6 Processing Time

To evaluate the bottlenecks and response time of the GeoCET algorithm, we evaluate the processing time of the principal components. In general, in the same geographical region, we compare the response times of different algorithms to discover the bottlenecks of the algorithm and the implementation that can optimize and improve.

Table 4 compares the processing time of two components on the same set of data sets. The first component of the measurement time overhead on a large proportion, factors affecting its performance include network access, routing policies, network bandwidth and so on, these times become significant. The processing time of the second component is stable and does not fluctuate much. Fast CPU and parallelization can reduce this time. The results in Table 5 show that our GeoCET algorithm outperforms the other two algorithms in terms of positioning response time.

Table 4. The processing time of GeoCET's components in different measurement datasets.

Different datasets	Delay-distance generation	Candidate landmark localization
Data in China	329 ms	345 ms
Data in Mumbai	548 ms	350 ms
Data in Silicon Valley	93 ms	300 ms
Data in Frankfurt	110 ms	329 ms

Table 5. The results of three algorithms in different measurement datasets.

Different datasets	GeoCET	Dragoon	Posit
Data in China	659 ms	892 ms	1,872 ms
Data in Mumbai	898 ms	1,201 ms	2,412 ms
Data in Silicon Valley	343 ms	980 ms	1,024 ms
Data in Frankfurt	510 ms	829 ms	1,548 ms

5 Discussion

As mentioned, It is clear that identifying the delay-distance model $\vec{f}_\rho$ and the nature of noise $\vec{\epsilon}$ is essential, as they have a profound impact on localization accuracy. According to the CRLB of a delay-distance model [7], their results show that the localization accuracy and the number of landmarks involved can be relevant, and their distance from the target cannot be too large. We used OWDs to alleviate the circuitousness of paths in the delay-distance model, and achieved the measurement work at the nearest city of the target nodes by leasing server nodes.

However, the proposed GeoCET method is limited: (1) The polynomial regression model and the elliptical trajectory constraints need to satisfy the characteristics of the 2-dimensional European space, i.e., the geometric structure (in the local environment of measurement) of the triangle inequality cannot be violated (TIV). (2) If the target node manipulates the response delay of the probe packet, it will forge the real position and deceive the geolocation algorithm. We have left this to future work.

We assumed that an IP geolocation method is to work on a global scale. It makes sense to understand GeoCET's coverage. Table 6 shows the coverage with the GeoCET algorithm in different regions. Across these four cities, GeoCET achieves more than 79.1% coverage.

Table 6. GeoCET's coverage in measurement data from different regions of the world.

Regions	Target IP nodes	GeoCET	Coverage
Beijing, Tianjin	1,247	963	77.2%
Mumbai	768	540	70.3%
Silicon Valley	576	496	86.1%
Frankfurt	693	600	86.6%

In China, our GeoCET identified 963 out of 1,247 target IP nodes, most IP nodes that cannot be located are due to (1) the delay measurement results violate the triangle inequality, (2) the PR model cannot describe the local delay-distance relationship, and (3) the negative constraint of the analysis node selection on the measurement. In Mumbai, GeoCET finds 540 out of 768 IP addresses for a 70.3% coverage. Of the ones that GeoCET missed, about 28 nodes did not respond to the probe packets, and the remaining nodes had a longer delay and were excluded as outliers. In Silicon Valley and Central Europe, the GeoCET can localize 86.1% and 86.6% the target IP addresses respectively. In addition to some unresponsive nodes, some of the target nodes are farther away from the analysis node than the initial threshold. We then manually inspected the remaining uncovered target nodes but not identified by GeoCET, tried and repeated multiple iterations in different periods or reselected the analysis nodes, and it turns out that our algorithm can solve more than 80% of the uncovering nodes.

Finally, we evaluated GeoCET's flexibility and scalability. The GeoCET is flexible enough to support extended functionality or improve its scalability. The GeoCET algorithm relies on fewer analysis nodes than other algorithms, which uses the intersection of the elliptical trajectories, its constraint is stronger than the triangulation method, and its stability is better on a real data set. For scalability, GeoCET can be stretched to larger areas, and many of its components can be parallelized. When the distribution of analysis nodes is sparse or the number of nodes is small, the localization accuracy and stability of GeoCET are better than similar algorithms. This phenomenon is also highlighted by Fig. 6b.

6 Conclusion

In this paper, we describe a novel accurate method for IP geolocation, namely, GeoCET. Our methodology estimates geographic location using elliptical trajectories intersection combined with maximum log-likelihood estimation technique. We also use a polynomial regression to fit the delay-distance model. It mitigates the effects of noisy distance estimation from measurements.

We assess the performance of GeoCET using four datasets of latency measurements collected from hundreds of nodes on the Internet where they are distributed in different regions of the world such as the China, India, Western United States, and Central Europe. Experimental results show that GeoCET can identify the geographic location of target nodes with a median error of 500–1000 m and 300–800 ms processing time. We compare it with implementations of the current existing measurement-based IP geolocation methodologies on the same datasets, and these results highlight the efficient performance of our approach and lower deployment costs.

As our future work, we will investigate better probability distributions for delay-distance data which can capture the behavior of noise in latency measurements. Then, we plan to extend the testing scope of the GeoCET method to cover more regions around the world to verify its scalability and stability.

Acknowledgment. We thank the anonymous reviewers whose comments helped improve the paper. We also thank the volunteers who provide the right location nodes. This work has been supported by the National Key Research and Development Program of China (grant no. 2016YFB0801300, 2016YFB0801304 and 2017YFB081701).

References

1. Apnic - query the apnic whois database. http://wq.apnic.net/apnic-bin/whois.pl
2. Maxmind: Detect online fraud and locate online visitors. http://www.hostip.info/
3. Skyhook: Location technology and intelligence. https://www.skyhookwireless.com/
4. Abdou, A., Matrawy, A., Van Oorschot, P.C.: CPV: delay-based location verification for the internet. IEEE Trans. Dependable Secure Comput. **14**(2), 130–144 (2017)
5. Allman, M., Beverly, R., Trammell, B.: Principles for measurability in protocol design. ACM SIGCOMM Comput. Commun. Rev. **47**(2), 2–12 (2017)

6. Bajpai, V., Eravuchira, S.J., Schönwälder, J.: Dissecting last-mile latency characteristics. ACM SIGCOMM Comput. Commun. Rev. **47**(5), 25–34 (2017)
7. Ciavarrini, G., Greco, M.S., Vecchio, A.: Geolocation of internet hosts: accuracy limits through cramér-rao lower bound. Comput. Networks **135**, 70–80 (2018)
8. Ciavarrini, G., Luconi, V., Vecchio, A.: Smartphone-based geolocation of internet hosts. Comput. Networks **116**, 22–32 (2017)
9. Dan, O., Parikh, V., Davison, B.D.: Improving IP geolocation using query logs. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 347–356. ACM (2016)
10. Ding, S., Luo, X., Yin, M., Liu, Y., Liu, F.: An IP geolocation method based on rich-connected sub-networks. In: 2015 17th International Conference on Advanced Communication Technology (ICACT), pp. 176–181. IEEE (2015)
11. Eriksson, B., Barford, P., Maggs, B., Nowak, R.: Posit: a lightweight approach for IP geolocation. ACM SIGMETRICS Perform. Eval. Rev. **40**(2), 2–11 (2012)
12. Eriksson, B., Barford, P., Sommers, J., Nowak, R.: A learning-based approach for IP geolocation. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 171–180. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12334-4_18
13. Gergonne, J.: The application of the method of least squares to the interpolation of sequences. Historia Mathematica **1**(4), 439–447 (1974)
14. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of internet hosts. IEEE/ACM Trans. Networking (TON) **14**(6), 1219–1232 (2006)
15. Hillmann, P., Stiemert, L., Rodosek, G.D., Rose, O.: Modelling of IP geolocation by use of latency measurements. In: 2015 11th International Conference on Network and Service Management (CNSM), pp. 173–177. IEEE (2015)
16. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards IP geolocation using delay and topology measurements. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 71–84. ACM (2006)
17. Laki, S., Mátray, P., Hága, P., Csabai, I., Vattay, G.: A model based approach for improving router geolocation. Comput. Networks **54**(9), 1490–1501 (2010)
18. Laki, S., Mátray, P., Hága, P., Sebők, T., Csabai, I., Vattay, G.: Spotter: a model based active geolocation service. In: 2011 Proceedings IEEE INFOCOM, pp. 3173–3181. IEEE (2011)
19. Li, D., et al.: IP-geolocation mapping for moderately connected internet regions. IEEE Trans. Parallel Distrib. Syst. **24**(2), 381–391 (2013)
20. Magee, L.: Nonlocal behavior in polynomial regressions. Am. Stat. **52**(1), 20–22 (1998)
21. Padmanabhan, V.N., Subramanian, L.: An investigation of geographic mapping techniques for internet hosts. In: ACM SIGCOMM Computer Communication Review, vol. 31, pp. 173–185. ACM (2001)
22. Percacci, R., Vespignani, A.: Scale-free behavior of the internet global performance. Eur. Phys. J. B-Condens. Matter Complex Syst. **32**(4), 411–414 (2003)
23. Shavitt, Y., Zilberman, N.: A geolocation databases study. IEEE J. Sel. Areas Commun. **29**(10), 2044–2056 (2011)
24. Shue, C.A., Paul, N., Taylor, C.R.: From an IP address to a street address: using wireless signals to locate a target. In: Proceedings of the 7th USENIX Conference on Offensive Technologies, p. 8. USENIX Association (2013)

25. Trammell, B., Kühlewind, M.: Revisiting the privacy implications of two-way internet latency data. In: Beverly, R., Smaragdakis, G., Feldmann, A. (eds.) PAM 2018. LNCS, vol. 10771, pp. 73–84. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76481-8_6
26. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: A comprehensive framework for the geolocalization of internet hosts. In: NSDI, vol. 7, p. 23 (2007)
27. Zhao, F., Luo, X., Gan, Y., Zu, S., Cheng, Q., Liu, F.: IP geolocation based on identification routers and local delay distribution similarity. Concurrency and Computation: Practice and Experience, p. e4722 (2018)
28. Ziviani, A., Fdida, S., de Rezende, J.F., Duarte, O.C.M.: Improving the accuracy of measurement-based geographic location of internet hosts. Comput. Networks **47**(4), 503–523 (2005)