

Network measurement based modeling and optimization for IP geolocation

Ziqian Dong^a, Rohan D.W. Perera^b, Rajarathnam Chandramouli^b, K.P. Subbalakshmi^b

^a*Department of Electrical and Computer Engineering
New York Institute of Technology
New York, NY 10023, USA
E-mail: ziqian.dong@nyit.edu*

^b*Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ 07030, USA
E-mail: {rperera, mouli, ksubbala}@stevens.edu*

Abstract

IP geolocation plays a critical role in location-aware network services and network security applications. Commercially deployed IP geolocation databases may provide outdated or incorrect location of Internet hosts due to slow record updates and dynamic IP address assignment by the ISPs. Measurement-based IP geolocation is used to provide real time location estimation of Internet hosts based on network delays. This paper proposes a measurement-based IP geolocation framework that provides location estimation of an Internet host in real time. The proposed framework models the relationship between measured network delays and geographic distances using segmented polynomial regression model and semidefinite programming for optimization. Weighted and non-weighted schemes are evaluated for location estimation. The proposed framework shows close to 17 and 26 miles median estimation error for nodes in North America and Europe, respectively. The proposed schemes achieve 70% to 80% improvement in median estimation error comparing to the first order regression approach for experimental data collected from Planet-Lab.

Keywords: IP geolocation, delay measurement, segmented polynomial, regression, Semidefinite programming.

1. Introduction

IP geolocation is the process of locating an Internet host or device that has an IP address. It plays a critical role in location-aware network services, such as targeted Internet advertising, content localization, restricting digital content sales to authorized jurisdictions, and security applications such as authenticating authorized users to avoid credit card fraud, locating suspects of cyber crimes and provide Internet forensic evidence for law enforcement agencies. An important application of IP geolocation is locating emergency calls initiated by voice over IP (VoIP) calls as mandated by the Federal Communications Commission (FCC) [1]. Statistics of the location information of Internet hosts or devices can also be used in network management and content distribution networks.

IP geolocation can be categorized into two types based on the technical approaches: database-based and measurement-based. Database-based IP geolocation has been widely used commercially. Companies like Akamai [2], Quova [3], Maxmind [4], Geobytes [5], Digital Envoy [6], etc. have maintained databases that associate IP addresses to geographic locations. A survey of IP geolocation techniques is presented [7]. These techniques include database-based technique such as *whois* database look-up, DNS LOC record, network topology hints on geographic information of nodes and routers, and measurement-based techniques such as round trip time (RTT) captured using *ping* and RTT captured via HTTP refresh.

Database-based IP geolocation methods rely on the accuracy of data in the database. The Internet service providers (ISPs) often use Dynamic Host Configuration Protocol (DHCP) to automatically assign IP addresses to a network host when it joins the IP network. A network host may be assigned different IP addresses at different times. The location records associated with IP addresses in the database may be obsolete or incorrect due to this dynamic IP address assignment and slow record updates. A commonly used database is the previously mentioned *whois* domain-based research services where a block of IP addresses is registered to an organization, and can be searched and located. These databases provide a rough location of the IP address. However, the information may be outdated and incomplete.

Measurement-based IP geolocation has been studied to use network delay or topology measurements to estimate geographic location of an Internet host [8–17]. A discussion of related work of measurement-based IP geolocation is presented in Section 2. The challenge of measurement-based IP geolocation approach is to find a proper model to represent the relationship between network delay measurements and geographic distances. Round-trip time (RTT) between network hosts, which is the time it takes for a packet to travel from a source to destination and then back to the source, is often used as a measure of network delay [18]. It is composed of propagation delay, transmission delay, processing delay and queueing delay. Propagation delay is considered as deterministic delay which is fixed for each path. Transmission delay, queueing delay and processing delay can be considered as stochastic delays, which can be modeled statistically. *traceroute* [19] and *ping* [20] are two network tools commonly used to measure RTTs. We use *traceroute* in our experiments to collect RTTs from Planet-Lab [21] nodes. Given RTT between network nodes, physical distance can be estimated using different curve fitting models. The geographic location of an IP can then be estimated using multilateration techniques based on measurements from several landmark nodes. Here, landmark nodes are defined as the Internet hosts whose geographic location is known. Previous works have considered linear regression model for IP geolocation. In this paper, we propose a measurement-based IP geolocation framework and test it with network delay measurements collected from Planet-Lab. However, hybrid methods that incorporate both measurement and database-based approaches can be considered to reduce execution time and achieve better accuracy.

The contributions of the proposed framework are listed below.

- A method of collecting and processing of real network data is discussed. The distribution of delay measurement for each chosen landmark node is analyzed. Noise removal technique is presented for data preparation.
- k -means clustering is applied to the dataset that groups measurement data into clusters with similar properties for each landmark node, where each region has a centroid that uses delay measurement and geographic distance as coordinates. We select landmark nodes close to the centroid in our framework to reduce the number of nodes required for taking delay measurements.
- A novel segmented polynomial regression model is proposed for mapping network delay to geographic distance for each landmark node. This approach gives fine granularity in defining relationship between the delay measurement and geographic distance.
- A convex optimization technique, semidefinite programming (SDP), is applied in finding the optimized solution for locating an IP given estimated distance from known landmark nodes.
- An integration of software tools, such as Matlab, Python and MySQL is implemented for the proposed IP geolocation framework.

The remainder of the paper is organized as the follows. Section 2 presents the related work. Section 3 introduces the proposed IP geolocation framework and detail description of each process. Section 4 presents the experimental results of the proposed methods using Planet-Lab dataset. Section 5 presents the conclusions.

2. Related Work

Methods of locating network hosts based on delay measurements have been studied in [8–17]. Some early work focused on network coordinate systems such as GNP [8], Virtual Landmarks [9], and Vivaldi [10], were done to evaluate network distance between Internet hosts. These techniques focus on network distance estimation which represents a topological distance in the network rather than the geographical distance. A systematic study of the IP-to-location mapping problem was presented in [11]. Geolocation tools such as *GeoTrack*, *Geoping* and *GeoCluster* were evaluated in this study.

The Cooperative Association for Internet Data Analysis (CAIDA) provides a collection of network data and tools for study on the Internet infrastructure [22]. *Gtrace*, a graphical traceroute provides a visualization tool to show the estimated physical location of an Internet host on a map [23]. A study on the impact of Internet routing policies to round trip times was presented in [24], where the problem posed by triangle inequality violations for the Internet coordinate systems. Placement of landmark nodes was studied in [25] to improve accuracy of geographic location estimation of a target Internet host.

Constraint-based IP geolocation (CBG) was proposed in [12] where the relationship between network delay and geographic distance is established using the bestline method using first-order linear regression and multilateration with distance constraint to estimate the geolocation of the target host. The experiment results show a 100 km median error distance for US dataset and 25 km median error distance for European dataset. However, the bestline method used in CBG does not consider the topology of the network which affects the geographic distance estimation. Topology-based geolocation method is introduced in [26]. This method extends the constraint multilateration techniques by using topology information to generate a richer set of constraints and applies optimization techniques to locate an IP. Geolocation using Buffering Delay estimation (GeoBud) was proposed in [27] where buffering delay at intermediate hops are considered in the CBG to improve estimation accuracy.

Octant is a framework proposed in [28] that considers both positive and negative constraints in determining the physical region of Internet hosts taken into consideration of the information of where the node can or cannot be. It uses Bézier-bounded regions to represent node position that reduces estimation region size. This method introduces a large amount of variants as both positive and negative constraints that increase the complexity of the framework.

Recent research interests focus on applying statistical tools and data mining technique in IP Geolocation. A statistical geolocation scheme of Internet hosts is proposed in [13]. The estimation of IP location is achieved by applying kernel density estimation to delay measurement and using maximum likelihood estimation of distance to landmarks. A combined gradient descent and forced-directed method is used for the estimation. A study on IP geolocation using maximum likelihood estimation technique is presented in [17] where both simulated data and real data are studied to validate the accuracy of the maximum likelihood technique. A method of using oneway delay constraints and path-latency model to locate routers is proposed in [14]. Geolocation techniques using text mining techniques on web contents and textual clues were proposed in [15] and [16].

Previous works have considered linear regression model for IP geolocation. In this paper, we propose a measurement-based IP geolocation framework that uses k -means clustering to cluster measurement data and apply segmented polynomial regression to model geographic distance based on network delay measurement and semidefinite programming to find the optimized location estimation of an IP address. The challenges in measurement-based IP geolocation include many factors. The path the packets take does not follow a straight line due to the circuitry of the network comparing to the point-to-point geographic distance measurement. Different network interfaces and processors render various processing delays. The uncertainty of network traffic makes the queueing delay at each router and host unpredictable. Therefore, a linear estimation of the relationship between network delay and geographic distance is not appropriate. This motivates us to explore geographic regions separately using segmented regression approach and model the geographic distance and delay measurement using segmented polynomials. The results show 70%-80% improvement in

median location estimation error with the segmented approach alone. Furthermore, IP spoofing and proxy usage can hide the real IP address. In our study, we assume the IP address of the Internet host is authentic, not spoofed or hidden behind proxies. To simplify notation, we refer to the host with IP address whose location is to be determined as IP in this paper.

3. Proposed IP Geolocation Framework

The objective of the proposed framework is to increase accuracy of the geographic location estimation of an IP based on the real-time network delay measurement from multiple landmark nodes. To study the characteristics of each landmark node, we collect delay measurements from the landmark nodes to a group of destination nodes. A novel approach of using segmented polynomial regression model for each landmark node is introduced to model the relationship between the network delay measurements and the geographic distances. We apply multilateration and semidefinite programming (a convex optimization method) to estimate the optimized location of an Internet host using estimated geographic distances from multiple landmark nodes. Fig. 1 shows the architecture of the proposed system. The proposed framework is composed of the following processes: data collection, data processing, data modeling and location optimization. Fig. 2 shows the flow chart of the processes. The following sections explain each process in details.

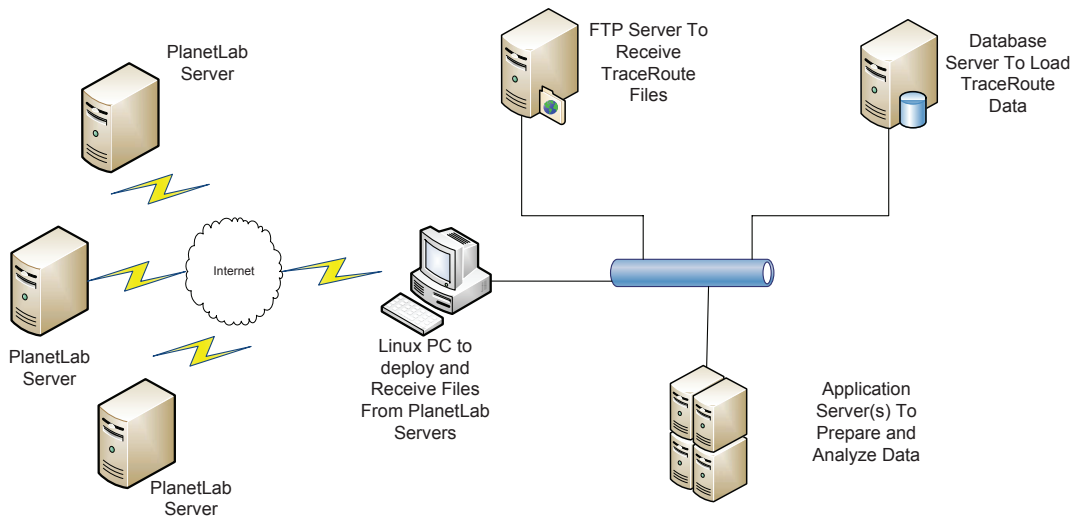


Figure 1: Measurement System Architecture.

3.1. Data Collection

We use Planet-Lab [21] for our network delay data collection. Planet-Lab is a global research network that supports the development of new network services. It consists of 1126 nodes at 517 sites around the globe. Planet-Lab requires all participants to provide their geographic locations, which gives a good

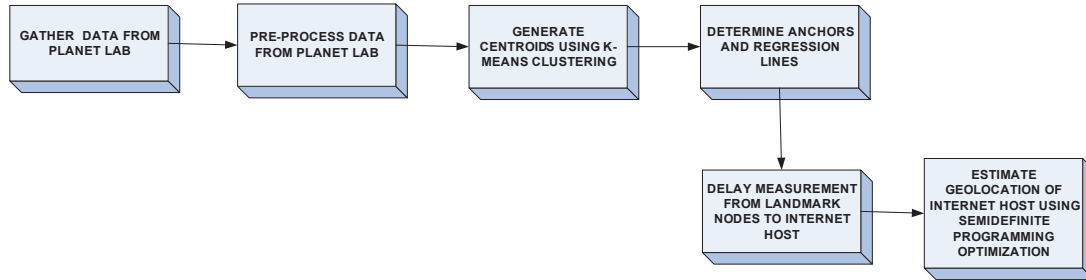


Figure 2: Flow chart of the proposed IP geolocation process.

reference to test the estimation errors of the proposed framework as the “Ground truth” of the actual node location is known. Due to the difference of maintenance schedules and other factors, Planet-Lab nodes are not accessible at all times. We selected 798 Planet-Lab nodes in our experiment. The selection of Planet Lab nodes is explained in Appendix A. We set up our experiment to take *traceroute* measurements every 60 seconds from the selected Planet-Lab nodes for a week during November 2010. We were able to collect data from 81 nodes from North America and 90 nodes from Europe which give consistent measurements as landmark nodes to initiate round-trip-time measurements to other Planet-Lab nodes. We use *traceroute* as our network delay measurement tool. However, other measurement tools can also be applied in our framework. The distribution of the selected nodes is shown in Fig. 3(a) for the North American nodes and Fig. 3(b) for the European nodes. Due to network blocking, we were not able to collect measurements from most South American and Asian Planet-Lab nodes.

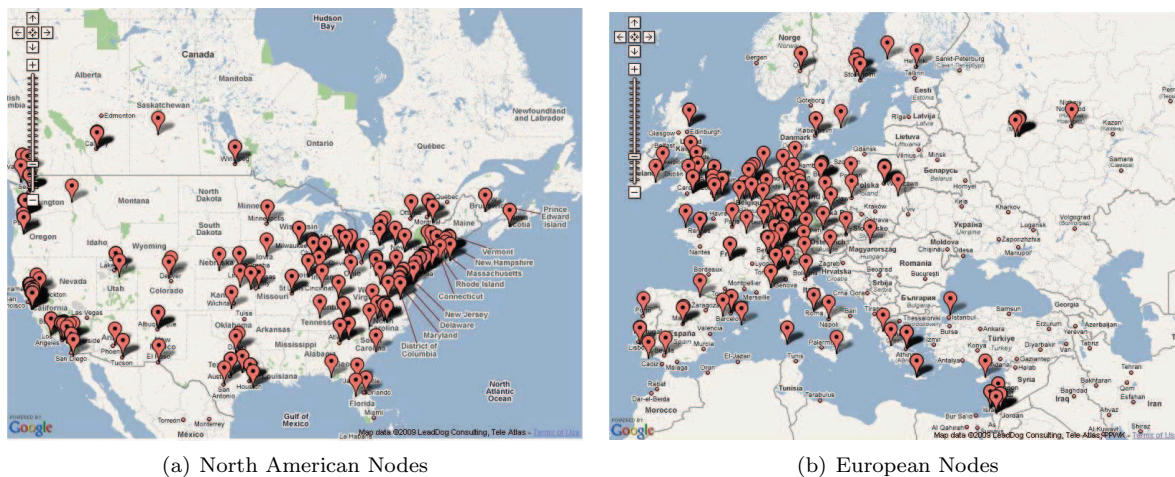


Figure 3: Distribution of selected Planet-Lab nodes

Delay measurements generated by *traceroute* are RTT measurements from a source node to a destination node. RTT is composed of propagation delay along the path, T_{prop} , transmission delay, T_{trans} , processing

delay, T_{proc} , and queueing delay, T_q , at intermediate routers/gateways. Processing delays in high-speed routers are typically in the order of microsecond or less. In our measurements, we observe RTT in the order of millisecond. Here processing delays are considered insignificant and are not considered. In this paper, RTT can be computed as the sum of propagation delay, transmission delay and queueing delay as shown in (1).

$$RTT = T_{prop} + T_{trans} + T_q. \quad (1)$$

Propagation delay is the time required for the energy of a signal to propagate from one point to another. It is considered as deterministic delay which is fixed for each path. A study shows that the speed of digital data travels along fiber optic cables is 2/3 the speed of light in a vacuum, c [29]. This sets an upper bound of the distance between two Internet nodes, given by $d_{max} = \frac{RTT}{2} \frac{2}{3} c$. Transmission delay is defined as the number of bits (N) transmitted divided by the transmission rate (R), $T_{trans.} = \frac{N}{R}$. The transmission rate is dependent on the link capacity and traffic load of each link along the path. Queueing delay is defined as the waiting time the packets experience at each intermediate router to be processed and transmitted. It is dependent on the traffic load at the router and processing power of the router. Transmission delay and queueing delay are considered as stochastic delays.

The challenges of data collection over the Internet through Planet-Lab nodes are: a) missing *traceroute* measurements due to the security settings at the intermediate routers, where *traceroute* maybe blocked. One example of the missing values in the measurements is shown in Fig.4 as marked in the square; b) Incomplete *traceroute* measurements when the path from one end node to another end node is blocked from probing packets. One such example is shown in Fig. 5 as marked in the square. These make about 73% of the our collected data unusable.

1	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	plgmu2.ite.gmu.edu	1.096	1.060	1.034			
2	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	FXHE-01-A-FR02.routers.gmu.edu	2.167	2.157	2.134			
3	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	matp-7609-1.v51.networkvirginia.net		13.966	14.009	14.047		
4	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	n1r13-router.networkvirginia.net		15.843	15.844	15.827		
5	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	newy-wash-98.layer3.nlr.net		23.935	23.939	23.919		
6	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	216.24.184.86		108.225	104.601	116.395		
7	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	as1.rtl.fra.de.geant2.net		123.690	112.010	112.001		
8	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	so-2-1-0.rtl.pra.cz.geant2.net		119.752	119.778	119.952		
9	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	so-2-0-0.rtl.bud.hu.geant2.net		128.582	128.605	128.643		
10	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	hungarnet-gw.rtl.bud.hu.geant2.net		125.405	129.863	129.835		
11	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	* 0 0 0						
12	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	c3550-vlan631.mol.hbone.hu		127.472	126.093	124.968		
13	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	* 0 0 0						
14	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	evghu.colbud.hu	121.036	121.905	121.509			
15	plgmu2.ite.gmu.edu	11/16/2010 05:54:04	evghu5.colbud.hu		121.593	121.582	119.009		

Figure 4: Traceroute result from Planet-Lab node *plgmu2.ite.gmu.edu* to *evghu5.colbud.hu*. Missing values at intermediate nodes.

3.2. Data Processing

To analyze the collected data, we first take a look at the distribution of the observed RTTs. At each landmark node, a set of RTT is measured for a group of destinations. Fig. 6(a) and 6(c) show the histograms

1	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	141.212.113.1	1.036	1.517	2.056			
2	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	141.213.127.134	0.582	0.567	0.566			
3	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	13-caen-btin-arb.r-btin-arbl.umnet.umich.edu		0.548	0.543	0.536		
4	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	v-btin-arbl-t2-wsu5.wsu5.mtch.net		1.631	1.625	1.768		
5	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	v0x1004.rtr.wash.net.internet2.edu		13.493	13.635	13.459		
6	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	abtlene-wash.rtl.fra.de.geant2.net		115.322	115.361	115.396		
7	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	so-6-2-0.rtl.vte.at.geant2.net		128.224	128.393	128.386		
8	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	as1.rtl.ath2.gr.geant2.net		156.142	156.187	156.279		
9	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	gmet-gw.rtl.ath2.gr.geant2.net		156.434	156.431	156.245		
10	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	ete1-to-ete2.backbone.gmet.gr		156.727	157.204	157.395		
11	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	lortissa2-to-ete1.backbone.gmet.gr		161.742	162.071	162.087		
12	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	clientRouter.uth.lortissa2.access-link.gmet.gr		162.093	161.037	161.114		
13	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
14	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
15	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
16	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
17	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
18	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
19	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			
20	planetlab5.eecs.umich.edu	11/16/2010 05:52:33	*	0	0	0			

Figure 5: Traceroute result from Planet-Lab node *planetlab5.eecs.umich.edu* to *iason.inf.uth.gr*. Missing values due to blocking.

of raw RTT measurements from three source nodes to their destined nodes in Planet-Lab. The unit of RTT measurement is millisecond, ms. It is shown in Fig. 6(a) that most of the RTT measurements fall between 10 ms and 15 ms with high frequency, while few measurements fall into the range between 40 ms to 50 ms with very low frequency. We treat the observations between 40 ms and 50 ms as outliers due to noisy data. Noisy measurement could be due to variation in network traffic that creates congestion on the path, therefore, resulting in longer delays. To reduce this noise, we apply outlier removal scheme to the raw measurement data.

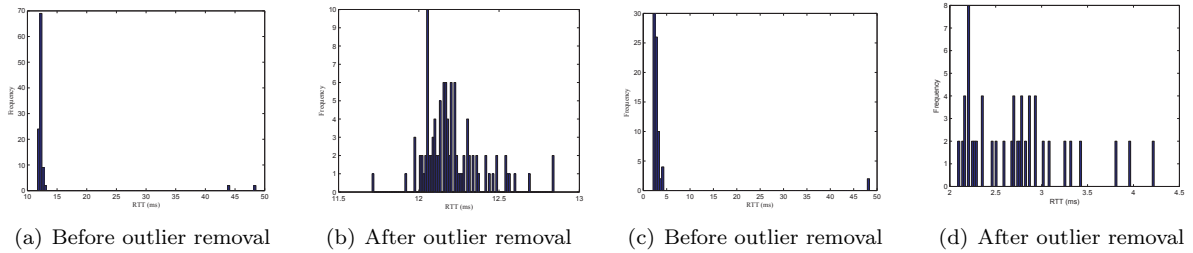


Figure 6: Histograms of RTT measurements from Planet-Lab nodes before a) c), and e) and after outlier removal b), d), and f).

The set of RTT measurements between the node i and node j is represented as T_{ij} , where $T_{ij} = \{t_1, t_2, \dots, t_n\}$, n is the number of measurements. We define the outliers as $t_i - \mu(T) > 2\sigma$, where $0 \leq i \leq n$. Here $\mu(T)$ is the mean of the set of data T and σ is the standard deviation of the observed data set. The data satisfy this condition is removed from the data set. The histogram after outlier removal is presented in Fig. 6(b) 6(d). Fig. 6(d) shows an example when RTT is short (within 10 ms). The RTT distribution tends to have high frequency on the lower end.

We group the data based on the RTT measurements and geographic distances for each landmark node into k clusters using k -means algorithm [30]. The algorithm is designed to solve the well known clustering problem, with objective of defining k centroids, one per cluster. The algorithm is composed of the following four steps:

1. Place k points into the space represented by the objects (RTT, distance) that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the k centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.

Here, the space is defined as distance vs. time. The objects are geographic distances between Planet-Lab nodes and the associated measured RTTs. Each cluster has a centroid with a set of value (RTT, distance) as coordinates. Fig. 7 shows an example of k -means clustering for data collected at Planet-Lab node *planetlab1.rutgers.edu* with $k = 5$. Each marker represents an observation of (RTT, distance) pair in the measurements. Different markers represent observations of (RTT, distance) pairs of different clusters. The notation “ \times ” represents the centroid of a cluster. This figure shows the observed data after outlier removal.

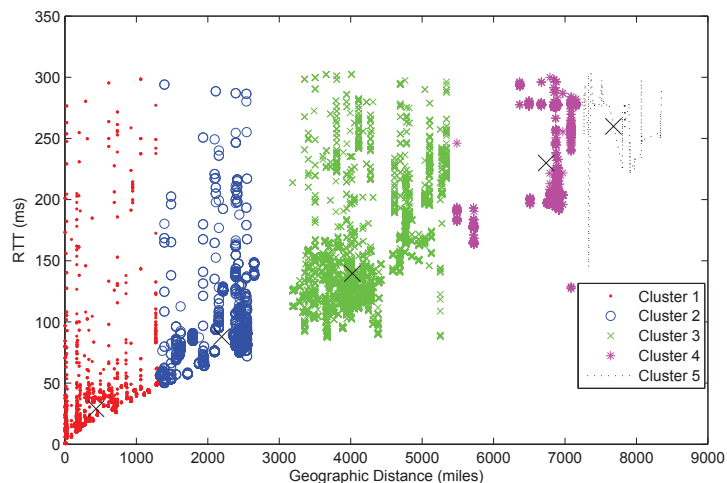


Figure 7: k -means clustering for collected data for Planet-Lab node *planetlab1.rutgers.edu*.

In k -means clustering process, we use $k=5$ as the number of clusters for each landmark node. Once a delay measurement is taken for an IP using random landmark selection, we estimate the region of the IP where the delay measurement will be mapped to one of the k clusters. Further measurements can be taken from the landmark nodes that are closer to the centroid of that cluster.

3.3. Segmented Polynomial Regression Model for Delay Measurements and Geographic Distance

The geographic distance of the Planet-Lab nodes where delay measurements are taken to the landmark node ranges from a few miles to 12,000 miles. Recent work [31] [11] use a least square fitting line to characterize the relationship between geographic distance, y , and network delay, x , where a and b are the first order coefficients, as shown in (2).

$$y = ax + b. \quad (2)$$

Due to different network set ups at different regions and non uniform distribution of the network nodes, the observed network delays show different characteristics for different regions. Linear regression model applied to the observed data from all regions may not be a good fit for characterizing network delay and geographic distance. We propose a regression model for the delay measurement vs. geographic distance for each landmark node based on regions with different distance ranges from the the landmark node. We call this regression model *segmented polynomial regression model* since the delay measurement is analyzed based on the range of distances to the landmark node. Fig. 8 shows an example of segmented regions around a landmark node using polynomial regression. After the data is clustered into k clusters for a landmark node, we segment the data into k groups based on distances to the landmark node. Cluster 1 (C_1) includes all delay measurements taken from nodes within R_1 radius of the landmark node, Cluster 2 (C_2) includes delay measurements between R_1 and R_2 , Cluster i (C_i) includes delay measurements between R_{i-1} and R_i . For each of the regions, we compute the polynomial coefficients to formulate the mapping of RTT to the geographic distance. This is done for every landmark node. Thus finer granularity can be achieved in the RTT vs. geographic distance model. The segmented polynomial regression to calculate distance y_k given network delay x_k for cluster k is shown as (3).

$$y_k = \sum_{i=0}^n a_n x_k^n, \quad x_k \in C_k, \quad (3)$$

where n is the order of the polynomials, a_n is the polynomial coefficient, C_k represents cluster k .

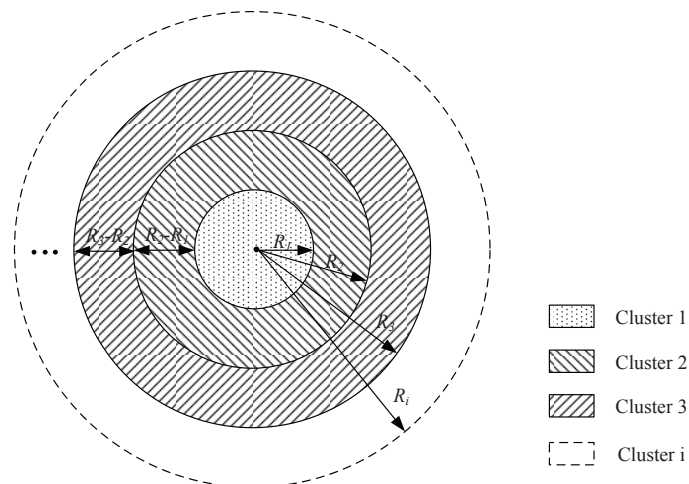


Figure 8: Example of segmented polynomial regression model for a landmark node.

First order regression analysis has been widely used in the study of relationship between geographic distance and network delay [11, 12, 31]. We studied different orders of regression lines in the proposed segmented polynomial regression model for each landmark node and found that lower order regression lines

provide better fit than higher order regression lines for our data set. Table 1 shows the coefficients of the segmented polynomial regression model for Planet-Lab node *planetlab3.csail.mit.edu*, where Table 2 shows its coefficients of the linear regression model. In our experiment, we evaluated polynomial orders from 1 to 10. The results show that the best fit polynomial order is 4 for our data set.

Fig. 9 shows the plot of the segmented polynomial in comparison with the first order linear regression approach for the same set of data. Since the landmark nodes are located with a non uniform distribution, where a large number of landmark nodes are located in densely populated areas and fewer nodes are located in less densely populated areas, there’s a gap between the polynomials of different segments. To decide the geographic distance based on the measurement RTT in regions that overlaps two regions, we take the average of the mapped geographic distance calculated using polynomials of both regions. For example, given an RTT measurement of 15ms from *planetlab3.csail.mit.edu*, we use polynomials for C_1 and C_2 to calculate the distance and use the mean of the two calculated distances as the estimated distance. The segmented polynomial regression models each geographical region separately comparing to non-segmented linear regression approaches. It provides tailored mapping of geographic distance to network delay for each geographical region. In Fig. 9, when RTT is small or the distance range is between 0 to 500 miles, the regression lines differ greatly between segmented regression-Region 1 and the linear regression line.

Table 1: Coefficients of segmented regression polynomials for Planet-Lab node *planetlab3.csail.mit.edu*.

Region	a_0	a_1	a_2	a_3	a_4
C_1	-0.000002	0.001579	-0.327457	20.946144	-15.044738
C_2	0	0.000223	-0.112349	10.955965	448.473577
C_3	-0.000065	0.02321	-2.836962	137.305958	-837.6261
C_4	0.000043	-0.018368	2.768478	-169.190563	5756.416625
C_5	-0.000006	0.004554	-1.152234	118.721352	-1839.132

Table 2: Coefficients of first order regression approach for Planet-Lab node *planetlab3.csail.mit.edu*.

Region	a_0	a_1
R	22.13668	402.596356

We will show the improved results of our proposed segmented polynomial regression versus first order linear regression approach in Section 4. The algorithm for the segmented polynomial generation is shown in Algorithm 1.

The process of locating an IP is as follows. When an IP is given for geolocation, a set of landmark nodes is randomly chosen to take delay measurement to the IP. Based on the measured delay from each landmark, the cluster of the IP can be defined. Landmarks that belong to the cluster will be chosen to take further delay measurements to the IP. Calculation of distance of the IP to the landmarks is done using the polynomials associated with that cluster. We apply semidefinite programming to find the optimized

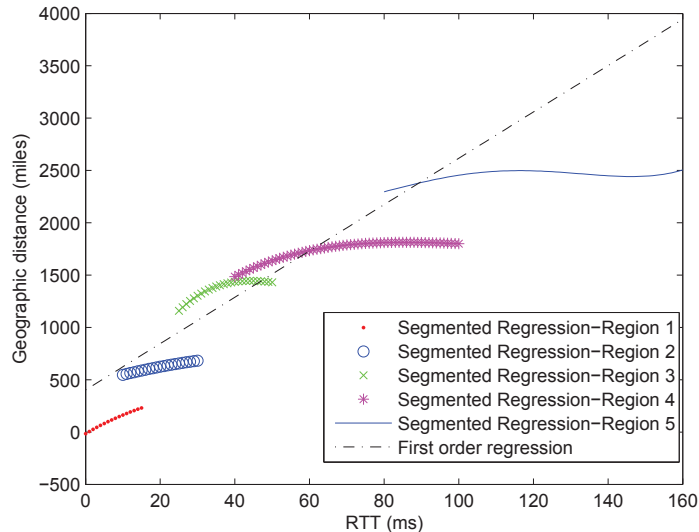


Figure 9: Example of segmented polynomial regression and first order linear regression for Planet-Lab node *planet-lab3.csail.mit.edu*.

estimated location of the IP, which is explained in Section 3.4.

3.4. Geolocation Estimation using Semidefinite Programming

Given estimated distances from landmark nodes to an IP, we use multilateration to estimate location of the IP. Multilateration is the process of locating an object based on the time difference of arrival of a signal emitted from the object to three or more receivers. This method has been applied in geolocation of Internet host in [12]. Fig. 10 shows an example of multilateration that uses three reference points L_1 , L_2 and L_3 to locate an Internet host, L_4 . In this example, round trip time to the Internet host L_4 with IP whose location is to be determined is measured from three Internet hosts with known locations L_1 , L_2 , and L_3 . Geographic distances from L_1 , L_2 , and L_3 to the L_4 are represented as d_{14} , d_{24} , and d_{34} , which is based on propagation delay. e_{14} , e_{24} , and e_{34} are additive delay from transmission and queueing delays. The radius of the solid circle shows the lower bound of estimated distance. The radius of dotted circle is estimated using a linear function of RTT [12]. The circle around each location shows the possible location of the IP. The overlapping region of the three circles indicates the location of the IP.

Due to circuitry of routing paths and variations of RTT measurement under different traffic scenario, it is difficult to find a good estimate between RTT and geographic distance. We apply the proposed segmented polynomial regression model explained in the previous subsection to represent the relationship between RTT and geographic distance to give fine granularity in modeling. We use this approach to map the mean measured RTT between node i to node j to a geographic distance, \hat{d}_{ij} . Semidefinite programming (SDP) algorithms have been studied to solve sensor network location problem [32]. We apply SDP to solve IP geolocation problem. The following notations are used in formulation of the optimization problem. We

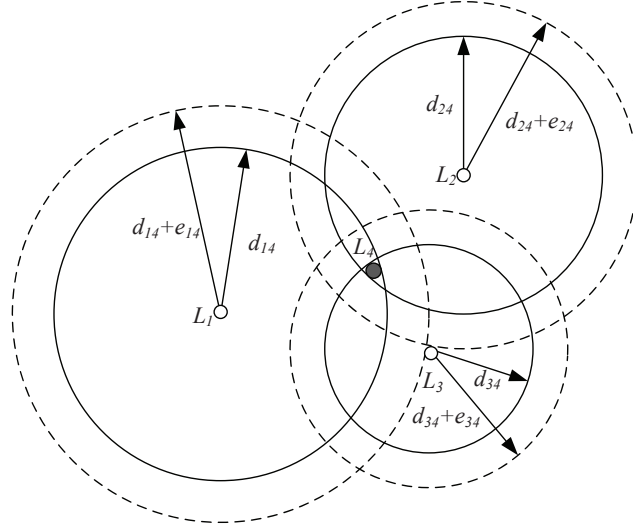


Figure 10: Multilateration for IP geolocation.

consider a network with N nodes, where m nodes are landmark nodes and n nodes are the network nodes with unknown location, where $N = m + n$. The coordinates of location of the landmark nodes is represented as a vector \mathbf{a}_k in a two-dimensional space \mathcal{R}^2 , $k = 1, \dots, m$, and the location of IP to be identified is represented as \mathbf{x}_i in \mathcal{R}^2 , $i = 1, \dots, n$. The actual geographic distance between two IPs with unknown locations \mathbf{x}_i and \mathbf{x}_j is denoted as d_{ij} . The actual geographic distance between an IP and a landmark node is d_{ik} . The estimated distance between nodes, whose locations are unknown, is denoted as $\|\mathbf{x}_i - \mathbf{x}_j\|$, $(i, j) \in \mathcal{N}$, where \mathcal{N} represents the set of nodes with unknown locations. The estimated distance between landmark nodes and nodes with unknown location is $\|\mathbf{x}_i - \mathbf{a}_k\|$, $(i, k) \in \mathcal{M}$, where \mathcal{M} represents the set of landmark nodes. $\mathcal{N} \cup \mathcal{M}$ define the set of nodes in the experiment. We evaluate different weights of the landmark nodes based on their distances to a centroid to study the effect of the placement of landmark nodes on estimation accuracy. γ_{ij} is the given weight defined as below. When $\gamma_{ij} = 1$, measurements from all landmarks are given equal weight. When $\gamma_{ij} = \frac{1}{d_{ij}}$, the weight is given in reverse proportion to distance of the landmark to the centroid. When $\gamma_{ij} = \frac{d_{ij}}{\sum d_{ij}}$, the weight is given based on the proportion of the distance of each landmark to the centroid over the total distance of all landmarks to the centroid.

$$\gamma_{ij} = \begin{cases} 1, & \text{equal weights for all landmark nodes;} \\ \frac{1}{d_{ij}}, & \text{reverse proportion to distance of landmark node to centroid;} \\ \frac{d_{ij}}{\sum d_{ij}}, & \text{proportion of the distance of one landmark node to centroid to all landmark nodes to centroid.} \end{cases}$$

The location estimation optimization problem can be formulated as a minimizing the mean square error

problem as in (4):

$$\min_{(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{R}^2} \left\{ \sum_{(i,j) \in \mathcal{N}} \gamma_{ij} |\|\mathbf{x}_i - \mathbf{x}_j\|^2 - d_{ij}^2| + \sum_{(i,k) \in \mathcal{M}} \gamma_{ik} |\|\mathbf{x}_i - \mathbf{a}_k\|^2 - d_{ik}^2| \right\}, \quad (4)$$

The matrix representation for coordinates of IPs with unknown locations is denoted as $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathcal{R}^{2 \times n}$. The matrix representation for coordinates for landmark nodes is denoted as $A = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m] \in \mathcal{R}^{2 \times m}$. e_i denotes the i^{th} unit vector in \mathcal{R}^n where only the i^{th} entry of the vector has value one and the rest are zeros.

The process of representing the problem in 4 using matrix representation in a space with both landmarks and nodes with unknown locations is explained as follows. The distance between two IPs with unknown locations can be represented using the following matrix representation $\|x_i - x_j\|^2 = e_{ij}^T X^T X e_{ij}$, where $e_{ij} = e_i - e_j$ is a vector in \mathcal{R}^n . The distance between an IP and the landmark node can be represented as $\|x_i - a_j\|^2 = a_{ij}^T [X, I_d]^T [X, I_d] a_{ij}$, where a_{ij} is the vector obtained by appending $-a_j$ to e_i in \mathcal{R}^n , I_d is the identity matrix in \mathcal{R}^m . Let $\mathcal{E} = \mathcal{N} \cup \mathcal{M}$, $Y = X^T X$, $g_{ij} = a_{ij}$ for $(i, j) \in \mathcal{M}$ and $g_{ij} = [e_{ij}; \mathbf{0}_d]$ for $(i, j) \in \mathcal{N}$. Equation 4 can be written in matrix form as:

$$\min_{(i,j) \in \mathcal{E}} \left\{ \gamma_{ij} |g_{ij}^T [Y, X^T; X, I_d] g_{ij} - d_{ij}^2| : Y = X^T X \right\}, \quad (5)$$

Problem (5) is not a convex optimization problem. To relax the problem to a convex optimization problem that can be solved by SDP, the constraint $Y = X^T X$ is relaxed to $Y \succeq X^T X$ [32]. Let $\mathcal{K} = Z : Z = [Y, X^T; X, I_d] \succeq 0$. The SDP relaxation of problem (5) can be written as SDP problem as in (6).

$$v^* := \min_{Z \in \mathcal{K}} \left\{ g(Z; D) := \sum_{(i,j) \in \mathcal{E}} \gamma_{ij} |g_{ij}^T Z g_{ij} - d_{ij}^2| \right\}. \quad (6)$$

To solve the above problem, we used **CVX**, a package for specifying and solving convex programs [33]. The computational complexity of SDP is analyzed in [32], which is bounded by $O(n^3)$, where n is the number of nodes whose locations are unknown and are to be estimated. In our case, we are locating one IP at a time, the computational complexity is limited to $O(1)$, where $n = 1$.

4. Experimental Results

The proposed framework is implemented in Matlab, Python and MySQL. We use python in our system because of its flexibility, well established interface with Matlab. We use **CVX** as the SDP solver. The regression polynomials for each landmark node was generated using our collected data from Planet-Lab. We tested our model using the Planet-Lab nodes as destined IPs. The mean RTT from landmark nodes to an IP is used as measured network delay to calculate distance. The estimated distance \hat{d}_{ij} is input to the SDP as distance

between landmark nodes and IP. The longitude and latitude of each landmark is mapped to a coordinate in \mathcal{R}^2 , which is the component of position matrix X . Cartesian coordinates are used to convert longitude and latitude of each geographic location to a two-dimensional representation [34]. Fig. 11 shows an example of the location of Planet-Lab node *planetlab1.rutgers.edu* calculated using SDP given delay measurements from a number of landmark nodes. The empty circle represents the location of the landmark nodes. The red circle represents the estimated location of the IP using SDP. The blue dot represents the actual location of the IP. The estimation error for this node is close to 10 miles.

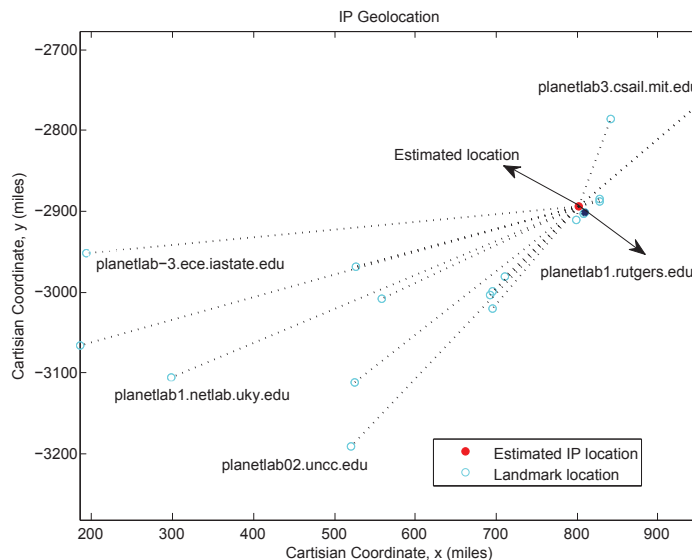
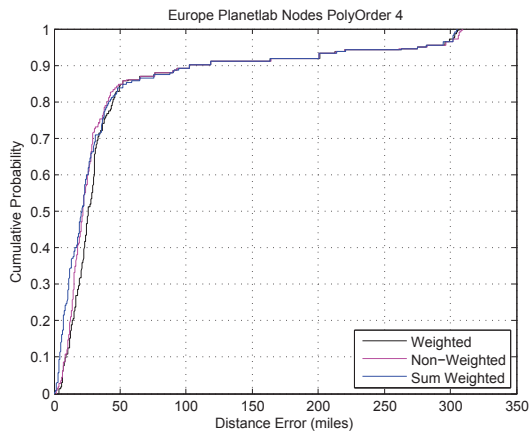


Figure 11: Screenshot of location estimation result of Planet-Lab node *planetlab1.rutgers.edu* using SDP approach.

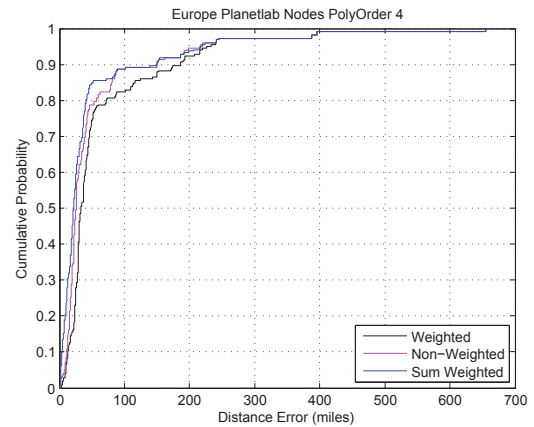
In this study, we show the results of locating a set of Planet-Lab nodes¹ given delay measurements from landmarks from Planet-Lab within a certain distance to the centroids to the Planet-Lab nodes. As the actual locations of the Planet-Lab nodes are provided, we can evaluate the estimation error of the proposed scheme comparing the estimated location with the actual location. Three schemes, namely non-weighted ($\gamma = 1$), weighted ($\gamma = 1/d_{ij}$) and sum weighted ($\gamma = d_{ij} / \sum d_{ij}$) are evaluated using SDP.

Fig. 12(a) and Fig. 12(b) show the empirical cumulative distribution function (CDF) of the estimation error in miles for European nodes using landmark nodes within 500 and 1000 miles to their centroids, respectively. Fig. 12(c) and Fig. 12(d) show the CDF of the distance error in miles for North American nodes using landmark nodes within 500 and 1000 miles to their centroids, respectively. The results show that weighted scheme that gives more weight to the landmarks that are closer to the centroid shows less estimation error than non-weighted and sum weighted schemes for the North American data set. The three schemes show similar performance with sum weighted scheme performs slightly better than the other two

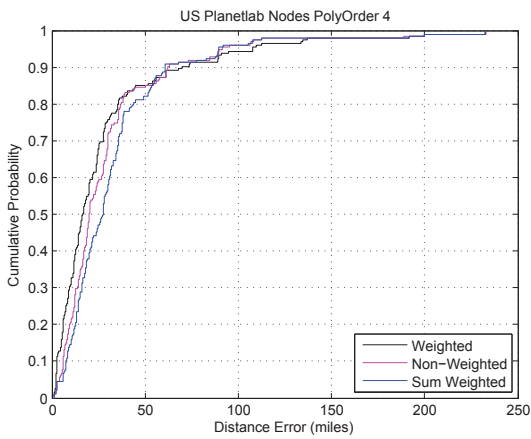
¹We choose the nodes from the list of nodes shown in [35] with distinct geographical locations.



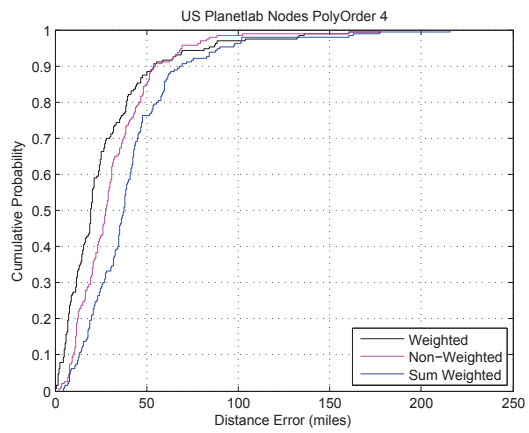
(a) European nodes using landmark nodes within 500 miles to centroid



(b) European nodes using landmark nodes within 1000 miles to centroid



(c) North American nodes using landmark nodes within 500 miles to centroid



(d) North American nodes using landmark nodes within 1000 miles to centroid

Figure 12: CDFs of estimation error of the proposed segmented regression of poly order 4.

schemes for the European data set. This is because the nodes are more concentrated in Europe and located in smaller regions than the nodes in North America. The median estimation error of the above experiments and non-segment linear regression using the same landmarks are summarized in Table 3. We also compare the results of segmented polynomial regression approach with order 1 polynomials (abbreviated as poly order 1 which represents linear regression) and order 4 polynomials (abbreviated as poly order 4) in Table 3. The proposed scheme has a 30.7 miles and 39.95 miles median estimation errors for European nodes using landmark nodes within 500 and 1000 miles to the associated centroids of the target IPs using poly order 1 in the segmented regression approach comparing to 25.7 miles and 33.0 miles estimation error using poly order 4 in the segmented regression approach. For North American nodes, the results are 19.0 miles and 21.6 using landmarks within 500 and 1000 miles to the associated centroids of the target IPs for poly order 1 and 16.8 and 19.6 for poly order 4. The improvement of accuracy in location estimation is shown with landmarks chosen closer to the centroids. Poly order 4 shows higher estimation accuracy than poly order 1 in the segmented regression approach. We also evaluate the estimation error with our proposed segmented polynomial regression with the non-segmented linear regression approach.

Table 3: Median Estimation Error (miles) Using Segmented Regression Model

Approach	Order	US (0-500 miles)	US (0-1000 miles)	Europe (0-500 miles)	Europe (0-1000 miles)
Segmented	1	19.0	21.6	30.7	39.9
Segmented	4	16.8	19.6	25.7	33.0
Non-segmented	1	98.9	113.5	110.7	222.8

Fig. 13 and Fig. 14 show the CDF comparison with the proposed segmented polynomial regression approach and the non-segmented linear regression approach for the North American nodes and European nodes, respectively. The median estimation error for non-segmented linear regression approach is shown in the third line item of Table 3. The results show up to 80% and 70% improvement in median estimation error by the proposed segmented polynomial regression approach than the non-segmented linear regression approach for North American nodes and European nodes, respectively. Because the node distribution in Planet-Lab shows concentration in geographical regions and the network set up in different countries and regions may vary. The segmented regression approach provides a more accurate modeling than the non-segmented regression approach.

5. Conclusions

We proposed a novel IP geolocation framework that incorporates k -means clustering, segmented polynomial regression modeling and semidefinite programming in network host geographic location estimation. The proposed segmented regression polynomial model for network delay and geographic distance clusters

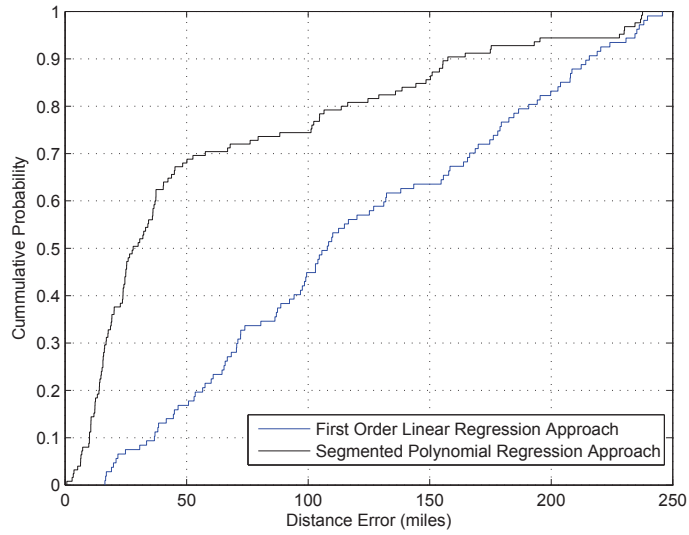


Figure 13: CDF of estimation error for North American nodes using segmented poly order 4 approach vs. linear regression approach.

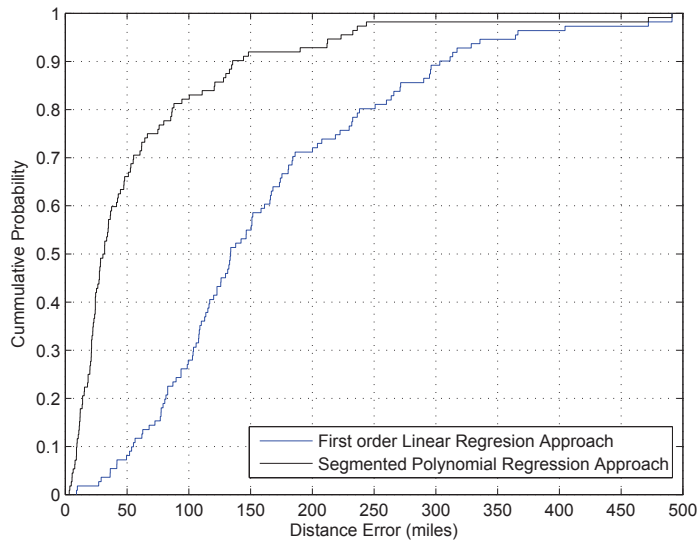


Figure 14: CDF of estimation error for European nodes using segmented poly order 4 approach vs. linear regression approach.

data into regions and models each region using an n^{th} order polynomial. This method allows finer granularity analysis for IP geolocation comparing to the conventional first order linear regression models. k -means clustering of the measured round trip delay data aims to group data into distinct regions. Semidefinite programming is applied to estimate the optimized location of an IP. Weighted and non-weighted schemes to consider the contribution of each selected landmark nodes are compared in the semidefinite programming. The median estimation error of our scheme is 17 and 26 miles for Planet-Lab nodes located in North American and Europe, respectively. The experimental results show up to 80% and 70% improvement in estimation error by our proposed segmented polynomial regression approach than the conventional first order linear regression approach for North American and European nodes, respectively. An average of 75% improvement in estimation error is achieved by the segmented regression model comparing to non-segmented regression model and an average of 13% improvement on the estimation error is archived by the segmented polynomial model comparing to the segmented linear model. The framework is implemented using open source softwares on Linux system. It can be implemented on network nodes running on Linux or Unix system. Due to network blocking and security settings at intermediate nodes, only 27% of the collected data were usable. This and other challenges, such as finding available landmark nodes to take delay measurements, implementing non-intrusive measurement tools, recovering missing delay measurements, and etc. remain in the measurement-based IP geolocation approaches. As network delays are highly dependent on the traffic load, it will be interesting to study the model for network delays and geographic distance under different load. As further work, it is of interest to study the queueing delay effects in the proposed model.

- [1] January 2011. [Online]. Available: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-05-116A1.pdf
- [2] "Akamai," 2008. [Online]. Available: <http://www.akamai.com>
- [3] "Quova," 2008. [Online]. Available: <http://www.quova.com>
- [4] "Maxmind," 2008. [Online]. Available: <http://www.maxmind.com>
- [5] "Geobytes," 2008. [Online]. Available: <http://www.geobytes.com>
- [6] "Digital envoy," 2008. [Online]. Available: <http://www.digitalenvoy.net>
- [7] J. A. Muir and P. van Oorschot, "Internet geolocation: Evasion and counterevasion," *ACM Computing Surveys*, vol. 4, no. 4, December 2009.
- [8] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," *IEEE INFOCOM*, June 2002.
- [9] L. Tang and M. Crovella, "Virtual landmarks for the internet," *ACM Internet Measurement Conf. 2003*, October 2003.
- [10] F. Dabek, R. Cox, F. Kaashoek, and R. Morris, "Vivaldi: A decentralized network coordinate system," *ACM SIGCOMM 2004*, August 2004.
- [11] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," *ACM SIGCOMM 2001*, August 2001.
- [12] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, December 2006.
- [13] I. Youn, B. L. Mark, and D. Richards, "Statistical geolocation of internet hosts," *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, pp. 1–6, August 2009.
- [14] S. Laki, P. Matray, P. Haga, I. Csabai, and G. Vattay, "A detailed path-latency model for router geolocation," *2009 5th*

International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, pp. 1–6, 2009.

- [15] C. Guo, Y. Liu, W. Shen, H. J. Wang, Q. Yu, and Y. Zhang, “Mining the web and the internet for accurate ip address geolocations,” *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pp. 2841–2845, April 2009.
- [16] C. Fink, C. Piatko, J. Mayfield, D. Chou, T. Finin, and J. Martineau, “The geolocation of web logs from textual clues,” *2009 International Conference on Computational Science and Engineering*, pp. 1088–1092, 2009.
- [17] M. J. Arif, S. Karunasekera, S. Kulkarni, A. Gunatilaka, and B. Ristic, “Internet Host Geolocation Using Maximum Likelihood Estimation Technique,” *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 422–429, 2010.
- [18] J. F. Kurose and K. W. Ross, *Computer Networking, A Top-Down Approach*, 5th ed. Addison Wesley, 2010.
- [19] “traceroute,” October 2008. [Online]. Available: <http://www.traceroute.org/>
- [20] “ping,” October 2008. [Online]. Available: <http://en.wikipedia.org/wiki/Ping>
- [21] “Planetlab,” 2008. [Online]. Available: <http://www.planet-lab.org>
- [22] “The cooperative association for internet data analysis,” November 2008. [Online]. Available: <http://www.caida.org>
- [23] “Gtrace,” November 2008. [Online]. Available: <http://www.caida.org/tools/visualization/gtrace/>
- [24] H. Zheng, E. K. Lua, M. Pias, and T. G. Griffin, “Internet routing policies and round-trip-times,” *Passive and Active Measurement Workshop (PAM 2005)*, March 2005.
- [25] A. Ziviani, S. Fdida, J. F. de Rezende, and O. C. M. B. Duarte, “Toward a measurement-based geographic location service,” *Passive and Active Measurement Workshop (PAM 2004)*, April 2004.
- [26] E. Katz-Bassett, J. John, A. Krishnamurthy, D. Weltherall, T. Anderson, and Y. Chawathe, “Towards IP geolocation using delay and topology measurements,” *Internet Measurement Conference 2008*, 2006.
- [27] B. Gueye, S. Uhlig, A. Ziviani, and S. Fdida, “Leveraging buffering delay estimation for geolocation of internet hosts,” *Networking 2006, 5th International IFIP-TC6 Networking Conference*, May 2006.
- [28] B. Wong, I. Stoyanov, and E. G. Sirer, “Octant: A comprehensive framework for the geolocalization of internet hosts,” *Proceedings of Symposium on Networked System Design and Implementation*, April 2007.
- [29] R. Percacci and A. Vespignani, “Scale-free behavior of the internet global performance,” *The European Physical Journal BCondensed Matter*, vol. 32, no. 4, April 2003.
- [30] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [31] A. Ziviani, S. Fdida, J. F. de Rezende, and O. C. M. B. Duarte, “Improving the accuracy of measurement-based geographic location of internet hosts,” *Computer Networks and ISDN Systems*, vol. 47, no. 4, March 2005.
- [32] P. Biswas, T. Liang, K. Toh, T. Wang, and Y. Ye, “Semidefinite programming based algorithms for sensor network localization,” *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 188–220, 2006.
- [33] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming (web page and software),” November 2008. [Online]. Available: <http://stanford.edu/~boyd/cvx>
- [34] “Coordinate geometry,” 2008. [Online]. Available: <http://mathworld.wolfram.com/SphericalCoordinates.html>
- [35] “Planet lab data,” 2011. [Online]. Available: <http://www.ece.stevens-tech.edu/~mouli/ipgeodata.tar.gz>

Appendix A. Planet Lab Node Selection

The selection of Planet Lab nodes is based on their availability and reliability at the time we took the measurements. Most of the nodes that are available are from the US and Europe. The Planet Lab experiment data is available at [35]. We deployed our script to gather round trip times (RTTs) from the

selected nodes to the list of 798 Planet Lab nodes. Due to network issues from some Planet Lab nodes in which the script was running we could only capture data from 206 (27%) of the nodes. Out of the 314 North America Nodes, we were able to take measurements from 81 nodes. Out of the 317 European Nodes, we were able to take measurements from 90 nodes.

The distribution of source nodes from other regions other than North America and Europe is as following

- North America 81
- Europe 90
- South America 14
- Asia 14
- Australia region (new zealand) 1
- Middle East Region (Israel) 6

```

SourceIP, MinParameterDistance, MaxParameterDistance, IncrementLevel, PolyOrder Error
StartIntervalDistance=MinParameterDistance
EndIntervalDistance=StartIntervalDistance+IncrementLevel
while EndIntervalDistance <= MaxParameterDistance do
  Retrieve Source LandMark By StartIntervalDistance, EndIntervalDistance and SourceIP
  if Source Landmark exists then
    Save LandMark, StartIntervalDistance, EndIntervalDistance, PolyOrder in Anchor Summary
    Table
    MinIntervalDistance=EndIntervalDistance
  else
    EndIntervalDistance=EndIntervalDistance+IncrementLevel
  end
end
end
foreach Landmark in Anchor Summary Table do
  if Regression Line DOES NOT Exist For Parameters
  (Landmark, StartIntervalDistance, EndIntervalDistance, PolyOrder) then
    Generate Regression Line For Above Parameters
  end
  Compute Estimated Distance Using Regression Line based on parameters in Anchor Summary
  Table
  if (Estimated Distance < MaxParameterDistance × 2) AND (Estimated Distance > 0) then
    Save Estimated Distance in File For Convex Optimization Routine
  end
  else
    Generate Regression Line For Source, MinParameterDistance, MaxParameterDistance
    Compute New Estimated Distance Using Regression Line based on parameters
    Source, MinParameterDistance, MaxParameterDistance
    if (New Estimated Distance < MaxParameterDistance × 2) AND (New Estimated Distance > 0)
    then
      Save Estimated Distance in File For Convex Optimization Routine
    end
  end
end
end
Determine SemiDefinite Optimization Based On Distance File

```

Algorithm 1: Segmented Polynomial Regression Algorithm