IP-Geolocation Mapping for Moderately Connected Internet Regions

Dan Li, *Member, IEEE*, Jiong Chen, Chuanxiong Guo, Yunxin Liu, *Member, IEEE*, Jinyu Zhang, Zhili Zhang, and Yongguang Zhang

Abstract—Most IP-geolocation mapping schemes [14], [16], [17], [18] take delay-measurement approach, based on the assumption of a strong correlation between networking delay and geographical distance between the targeted client and the landmarks. In this paper, however, we investigate a large region of moderately connected Internet and find the delay-distance correlation is weak. But we discover a more probable rule—with high probability the shortest delay comes from the closest distance. Based on this closest-shortest rule, we develop a simple and novel IP-geolocation mapping scheme for moderately connected Internet regions, called *GeoGet*. In GeoGet, we take a large number of webservers as passive landmarks and map a targeted client to the geolocation of the landmark that has the shortest delay. We further use JavaScript at targeted clients to generate HTTP/Get probing for delay measurement. To control the measurement cost, we adopt a multistep probing method to refine the geolocation of a targeted client, finally to city level. The evaluation results show that when probing about 100 landmarks, GeoGet correctly maps 35.4 percent clients to city level, which outperforms current schemes such as GeoLim [16] and GeoPing [14] by 270 and 239 percent, respectively, and the median error distance in GeoGet is around 120 km, outperforming GeoLim and GeoPing by 37 and 70 percent, respectively.

Index Terms—IP geolocation, GeoGet, moderately connected Internet

1 Introduction

Many applications will benefit from or be enabled by knowing the geographical locations (or geolocations) of Internet hosts. Such locality-aware applications include local weather forecast, the choice of language to display on webpages, targeted advertisement, page hit account in different places, restricted content delivery according to local policies, etc. Locality-aware peer selection will also help P2P applications in bringing better user experience as well as reducing networking traffic [1], [2], [3], [4].

Traditional IP-geolocation mapping schemes [14], [16], [17], [18] are primarily delay-measurement based. In these schemes, there are a number of landmarks with known geolocations. The delays from a targeted client to the landmarks are measured, and the targeted client is mapped to a geolocation inferred from the measured delays. However, most of the schemes are based on the assumption of a linear correlation between networking delay and the physical distance between targeted client and landmark. The strong correlation has been verified in some regions of the Internet, such as North America and Western Europe [14], [15]. But as pointed out in the literature [15], the Internet connectivity around the world is very complex, and such strong correlation may not hold for the Internet everywhere.

In this paper, we investigate the delay-distance relationship in a particular large region of the Internet (China), where the Internet connectivity is moderate. The data set contains hundreds of thousands of (delay, distance) pairs collected from thousands of widely spread hosts. We have two observations from the data set. First, the linearity between the delay and distance in this region of Internet is positive but very weak. Second, with high probability the shortest delay comes from the closest distance, and we call this phenomenon the "*closest-shortest*" rule.

Based on the observations, we develop a simple yet novel IP-geolocation mapping scheme for moderately connected Internet regions, called GeoGet. In GeoGet, we map the targeted client to the geolocation of the landmark that has the shortest delay. We take a large number of webservers with wide coverage and known geolocations as passive landmarks, which eliminates the deploying cost of active landmarks. We further use JavaScript at targeted clients to generate HTTP/Get probing for delay measurement, eliminating the need to install client-side software. To control the measurement cost, we step-by-step refine the geolocation of a targeted client, down to city level. In practice, GeoGet can be deployed in combination with a certain locality-aware application such that the application can easily obtain the geolocations of their clients.

We implement GeoGet in the moderately connected Internet region we study (China). In the implementation, we collect a large number of webservers and choose about

- *D. Li is with the Department of Computer Science, Tsinghua University, Room 9-402, East Main Building, Beijing 100084, China. E-mail: tolidan@gmail.com.*
- *J. Chen and J. Zhang are with the Department of Computer Science, Peking University, Beijing 100083, China. E-mail: cjpickup@gmail.com, zjy@net.pku.edu.cn.*
- *C. Guo, Y. Liu, and Y. Zhang are with the Wireless and Networking Group, Microsoft Research Asia, Zhichun Road 49#, Beijing 100190. E-mail: {chguo, yunxin.liu, ygz}@microsoft.com.*
- *Z. Zhang is with the Department of Computer Science and Engineering, University of Minnesota, 4-192 Keller Hall, 200 Union Street SE, Minneapolis, MN 55416. E-mail: zhzhang@cs.umn.edu.*

Manuscript received 4 Sept. 2011; revised 7 Feb. 2012; accepted 19 Apr. 2012; published online 2 May 2012.
Recommended for acceptance by Y.C. Hu.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-09-0593.
Digital Object Identifier no. 10.1109/TPDS.2012.136.

40,000 of them as passive landmarks, whose geolocations can be accurately obtained. The passive landmarks cover the entire region we are interested. We deploy a coordination server in combination of a website providing video-on-demand (VOD) service, and attract more than 5,000 clients from diverse geolocations to visit and participate during our measurement interval.

The evaluation results show that when probing about 100 landmarks, GeoGet accurately maps 35.4 percent targeted clients to city level, which outperforms existing schemes such as GeoLim [16] and GeoPing [14] by 270 and 239 percent, respectively, and the median error distance in terms of city in GeoGet is around 120 km, outperforming GeoLim and GeoPing by about 37 and 70 percent, respectively.

The contributions of this paper are twofold. First, by studying a large data set, we show that most of the traditional IP-Geolocation mapping schemes cannot work well for moderately connected Internet regions, since the linear delay-distance correlation is weak in this kind of Internet regions. Second, based on the measurement results (MR), we develop and implement GeoGet, which uses the closest-shortest rule and works much better than traditional schemes in moderately connected Internet regions. We acknowledge that we are not the first to apply the closest-shortest rule and the mapping accuracy of GeoGet is still not very high. However, we go a large step toward developing a better IP-Geolocation system for moderately connected Internet regions. We believe the accuracy will improve significantly if probing more landmarks.

The rest of this paper is organized as follows: Section 2 introduces the related work. Section 3 investigates the delay-distance relationship in the region we study. Section 4 develops GeoGet based on the closest-shortest rule. Section 5 discusses the implementation and Section 6 shows the evaluation results. Section 7 presents the conclusion.

2 Related Work

Delay-measurement approach. Various schemes have been proposed for IP-geolocation mapping, and most of them take delay-measurement approach [14], [16], [17], [18], [23], [24], [25], [26], [27]. In this approach, there are landmarks with known geolocations, and the networking delays between a targeted client and landmarks are measured. The geolocation of the targeted client is inferred from the measured results. In what follows, we introduce some representative schemes taking this approach, including GeoPing [14], GeoLim [16], TBG [17], and Octant [18].

In GeoPing [14], there are a number of landmarks and probing hosts (in practice, the landmarks and probing hosts are usually overlapped and thus the landmarks are *active landmarks*). Each probing host uses ICMP probing to measure its delays to a targeted client as well as all the landmarks. As a result, every landmark and the targeted client get a delay vector to all the probing hosts. Then, the geolocation of the targeted client is mapped to the location of the landmark whose delay vector has the shortest euclidean distance with that of the targeted client. Therefore, the mapping accuracy of GeoPing depends on strong delay-distance correlation, since it maps the similarity of vectors in distance dimension to that in delay dimension.

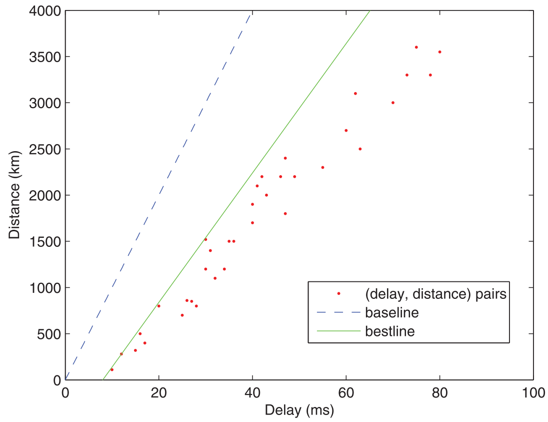


Fig. 1. Illustration of GeoLim. Baseline is drawn by the ideal digital transmitting speed in fiber. Bestline is fit tightly above all measured (delay, distance) pairs, and it is used for distance extraction for a delay value.

They find that such strong correlation holds at least for richly connected Internet regions such as North America. But for Internet regions where delay-distance correlation is weak, this mapping between delay dimension and distance dimension will introduce large error.

GeoLim [16] uses distance constrains based on measured delays to geolocalize a targeted client. Each landmark first measures its delays to the other landmarks, and fits a *bestline* tightly above all the (delay, distance) pairs measured, as shown in Fig. 1. There is also a *baseline*, which is drawn by the ideal digital transmitting speed in fiber (2/3 of the light speed), and certainly it lies above the bestline. Given the delay measured from a landmark to the targeted client, the landmark extracts the distance from the delay value based on the bestline, and draws a circle with its own geolocation as the center and the extracted distance as the radius. If all the circles drawn by the landmarks intersect to a region, the centroid of the region is regarded as the geolocation of the targeted client. In fact, GeoLim also assumes a moderate or strong delay-distance correlation. Otherwise, the extracted distance based on the bestline will be overly skewed compared with the actual distance, and consequently the mapping accuracy will degrade.

Katz-Bassett et al. [17] argue that the assumption on strong delay-distance correlation is unreliable when the delay (distance) is large. They propose to use network topology information to improve the mapping accuracy when there is no landmark with short delay (distance), and they call the scheme TBG. With *traceroute* tool, they first find the routers along the path from a deployed landmark to a targeted client and then use delay measurement to geolocalize the intermediate routers as well as the targeted client. TBG uses the maximum transmission speed of packets in fiber to calculate the distance constraint from the measured delay, and relies on global optimization to minimize the average error distance for the routers and targeted client. However, similar to GeoLim, when the delay-distance correlation is weak, the extracted distance from a measured delay value will be much overestimated. In addition, the global optimization may introduce extra errors for deciding the geolocation of the targeted client in an effort to reduce the errors to geolocalize the intermediate routers.

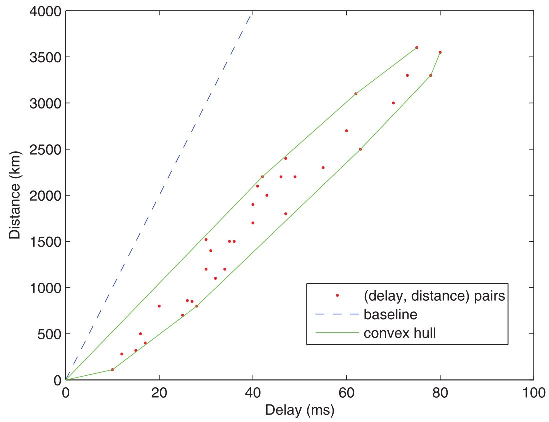


Fig. 2. Illustration of Octant. Baseline is drawn by the ideal digital transmitting speed in fiber. Convex hull is drawn as the upper bound distance constraint as well as lower bound distance constraint for a delay value.

Wong et. al. [18] bring forward Octant, which maps a targeted client to a geolocation region by use of not only positive constraints (where the targeted client might lie), but also negative constraints (where the targeted client cannot lie). The positive constraints indicate the upper bound distance of the targeted client, while the negative constraints indicate the lower bound distance. They formulate the IP-geolocation mapping problem as one error-minimizing constraint satisfaction, and solve the constraint system geometrically to yield the geolocation of the targeted client. Fig. 2 shows the convex hull to compute the upper bound distance and lower bound distance given the delay from a landmark to the targeted client. However, based on the data set, we study in the Internet regions where delay-distance correlation is weak, the empty lower right region in Fig. 2 does not exist. Octant also depends on delay-distance correlation to get reliable distance constraints from a measured delay.

Probably, the most related work with our work is [28]. Though bearing different design goals, the two works take similar approach, i.e., using webservers as landmarks and mapping the geolocation target to the closest landmark. However, our work differs from [28] in two aspects. First, we validate the weak linearity between delay and distance by a large data set from a moderately connected Internet region. Second, study in [28] requires Traceroute to infer the closest landmark but GeoGet uses client-side javascript, because the Ping/Traceroute commands are usually prohibited by many intermediate routers or webservers. In one sentence, our work for the first time focuses on the IP-Geolocation accuracy in moderately connected Internet regions, including measuring and mining the real data, validating the hidden relationship between delay and distance, and developing the real system.

Other approaches. There are also IP-geolocation mapping schemes that do not take delay-measurement approach. A simple and straightforward approach is to let end users manually input their geolocations. However, inaccuracy, inconsistency and incompleteness are unavoidable in this manual approach. NetGeo [8] and IP2LL [7] extract location information from the querying results returned by WHOIS database [6], which is maintained by ICANN/IANA [13]. The major problem of using WHOIS database is that the location information contained in WHOIS database may be outdated or incorrect. The location information in WHOIS database often indicates the address of the head office of the owner of an IP address block and may not be geographically related to the location of the corresponding IP addresses. A large IP address block owned by a large ISP contains many individual IP addresses which may be used by many hosts in different locations, but WHOIS database returns only one record for the whole block. In addition, many commercial companies like Quova [9] also provide IP-geolocation mapping services, but we are not aware of their technical details. Lately, some work argued against the accuracy of IP-geolocation mapping databases [22]. Hence, in the evaluation part, we use the geolocations that are agreed by three different databases as the ground truth.

In [20], the authors propose a new framework for IP geolocation which is reduced to a machine-learning classification problem. They consider a set of lightweight measurements from a set of known monitors to a target, and then classifies the location of that target based on the most probable geographic region given probability densities learned from a training set.

In [21], the webpages are mined to learn the geoloations of IP addresses. A three-stage approach is used to step-by-step infer the IP addresses of the web URLs by analyzing the *location*-related keywords in the web content. However, the IP addresses of the webservers and their subnets only cover a small subset of the global IP address space. In GeoGet, we seek to geolocalize the IP addresses of web clients, the information of which is much more important for location-aware applications. But the geolocations of the webservers obtained in [21] can help play as the landmarks in GeoGet.

3 Delay-Distance Relationship

There are many researches and discussions on delay-distance relationship. Padmanabhan and Subramanian [14] study the data sets in North America, and find that there is a strong delay-distance correlation. Ziviani et al. [15] also find that delay-distance correlation is strong within North America and within Western Europe, but weak for the entire Internet as a whole. As presented in the section above, most previous work on IP-geolocation mapping is based on the assumption of a strong delay-distance correlation. In this section, we investigate the delay-distance relationship from a large data set collected in a particular region of the Internet, China, which is the world's largest country in terms of the number of Internet users and the second largest in terms of the size of IP address space [5]. To be consistent with prior work, we use *round-trip delay* (RT Delay) as the delay measurement in this paper.

3.1 Data Set

Our data set is composed of (delay, distance) pairs collected between 240 probing hosts and 6,000 webserver landmarks. Each (delay, distance) pair is unique for a (probing host, landmark) pair. The probing hosts and landmarks come from diverse geolocations in China. The values of distance

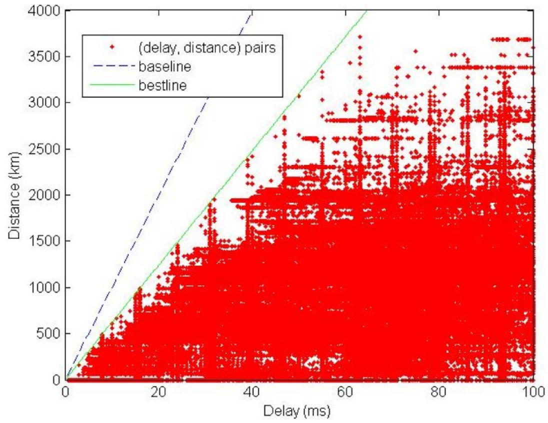


Fig. 3. (Delay, distance) pairs of our data set. Baseline is drawn by the ideal digital transmitting speed in fiber. Bestline is fit tightly above all measured (delay, distance) pairs.

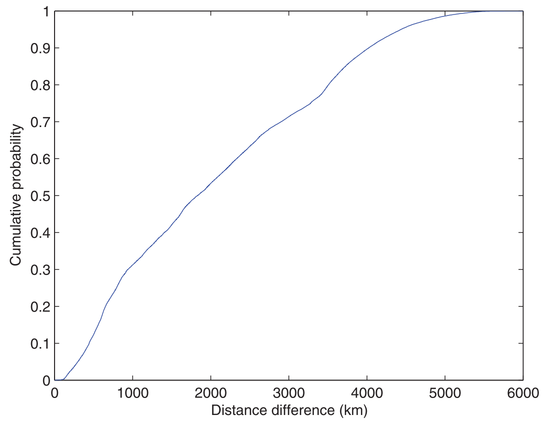


Fig. 4. CDF of the difference between extracted distance and actual distance if using bestline to extract the distance from delay.

and delay are obtained as follows: First, based on the known geolocations of the probing hosts and landmarks, we calculate the geographical distances of (probing host, landmark) pairs using Vincenty's formula [19], which assumes that the figure of the Earth is an oblate spheroid, and accordingly is more accurate than models assuming a spherical Earth. Second, the delays of (probing host, landmark) pairs are measured using HTTP request.[1] Each probing host measures the delay to a landmark for 10 times, and takes the minimum value as the delay between them. *Not all* probing hosts finish measuring all landmarks because of server failure, timeouts, or poor network connectivity. We only choose the (probing host, landmark) pairs whose minimum delays are within 100 ms (larger value is meaningless). Totally, we get 200,796 such (delay, distance) pairs in our data set. We believe this large data set represents the characteristic of the delay-distance relationship in China.

In the data set, the delay varies from 1 to 100 ms, with the median value of 52 ms, and the distance varies from 0 to 3,712 km, with the median value of 887 km. There are no biased delay values or distance values that have much higher percentage than others.

We plot all the (delay, distance) pairs in Fig. 3. As aforementioned, the baseline is drawn by the ideal digital transmitting speed in fiber (2/3 of the light speed), and the bestline is fit tightly above all (delay, distance) pairs. From this figure, we can observe that there is no obvious delay-distance correlation and the distance/delay slope for the bestline is about 64 km/ms. Further, there is no empty region in the lower right part (compared with that in Fig. 2), indicating that there is no reliable nonzero lower bound distance for a delay value.

We further study the delay-distance relationship in the data set, including *delay-distance correlation* and *closest-shortest rule*.

3.2 Delay-Distance Correlation

We use correlation coefficient to quantify the linearity of delay-distance correlation. Given all the delay values and distance values in the data set, assume the deviation of delay is $D(d_e)$, the deviation of distance is $D(d_i)$, and the covariance between delay and distance is $cov(d_e, d_i)$, then the correlation coefficient of delay and distance, $corr(d_e, d_i)$, is calculated as follows:

$$corr(d_e, d_i) = cov(d_e, d_i)/(sqrt(D(d_e)) * sqrt(D(d_i))).$$

The absolute value of $corr(d_e, d_i)$ is within [0, 1]. If that value is closer to 1, delay and distance are more strongly correlated.

The delay-distance correlation coefficient differs a lot from different regions of the Internet. Ziviani et al. [15] calculate that the delay-distance correlation coefficient is high for North America and Western Europe, from about 0.73 to 0.89; but the value for worldwide Internet is very low, around 0.3.

In our data set, the delay-distance correlation coefficient of all the (delay, distance) pairs is 0.3734. Therefore, the linearity between delay and distance is positive but rather weak in the Internet region we study. There are many reasons for the weak delay-distance correlation, such as networking congestion, circuitous paths, moderate inter-AS connection as well as unevenness in Internet connectivity within the region. Most of the reasons are owing to the moderate Internet connectivity.

For moderately connected Internet regions, if we use the bestline to extract the distance from a delay value, the result will be overly skewed for many cases. Fig. 4 shows the CDF of the difference between extracted distance and actual distance. The average for all (delay, distance) pairs is 2,152 km. This difference is too big considering the maximum distance geographical distance is only 3,712 km.

Therefore, though traditional IP-geolocation mapping schemes that depend on strong delay-distance correlation work well for richly connected Internet regions, they are not suitable for moderately connected Internet regions, for which we will further evaluate in Section 6.

3.3 Closest-Shortest Rule

We now study another property of delay-distance relationship—whether the shortest delay comes from the closest distance. We call this property *closest-shortest rule* and use the metric *dist-rank-of-shortest-delay* to evaluate it. Though

1. We will demonstrate later in Section 4 that HTTP request is a feasible method to measure the networking delay.

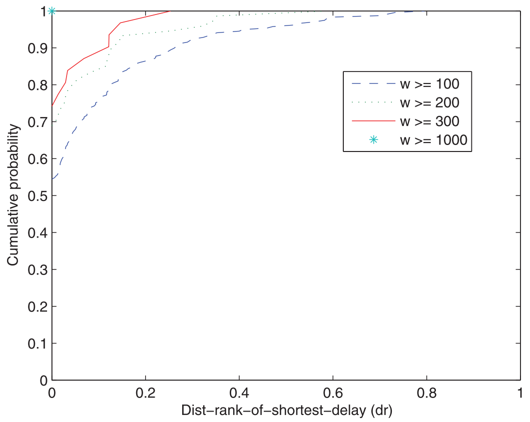


Fig. 5. CDF of dist-rank-of-shortest-delay for probing hosts that measures w landmarks. For $w \geq 1{,}000$, 100 percent of probing hosts have 0-value dist-rank-of-shortest-delay.

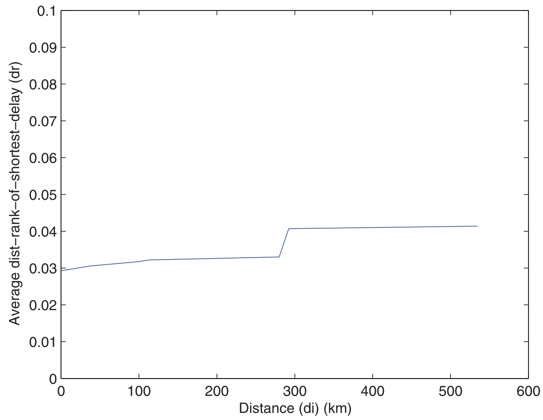


Fig. 6. Average dist-rank-of-shortest-delay when the distance to the closest landmark is within di.

there are previous works enclosing the same phenomenon, we are the first time to validate and quantify it in the moderately connected Internet regions.

In our data set, probing hosts measure the delays to different numbers of landmarks. For a specific probing host which measures w landmarks, there are w such (delay, distance) pairs. We choose the pair with the *shortest* delay from the w pairs, say, $(delay0, distance0)$. All the w distances are sorted from smallest to largest, and the ranking position (starting from 0) of $distance0$ is computed, say, r. The value of $\frac{r}{w}$ is defined as the *dist-rank-of-shortest-delay*, denoted as d_r.

Two factors affecting the value of d_r are the number of landmarks probed and the distance to the closest landmark. To study the impact of the number of landmarks, we select all the probing hosts with $w \geq 100$, $w \geq 200$, $w \geq 300$, and $w \geq 1{,}000$, and calculate the corresponding CDF of d_r. The result is shown in Fig. 5.

From this figure, d_r decreases with the growth of the number of landmarks measured, indicating that the closest-shortest rule is more likely to hold when measuring more landmarks. For those probing hosts with $w \geq 100$, 54.4 percent of them get the shortest delay from the closest distance, and 90 percent get the shortest delay from a distance ranking lower than 26.0 percent. For the probing hosts with $w \geq 300$, 74.2 percent get the shortest delay from the closest distance, and 90 percent get the shortest delay from a distance ranking lower than 12.1 percent. When the number of probed landmarks is sufficient, say, more than 1,000, 100 percent of the probing hosts have 0-value d_r. It means that the closest-shortest rule always holds in this case. The reason for the increase of the probability of $d_r = 0$ when measuring more landmarks is that, when a probing host measures more landmarks, it is more likely to find a closest server that returns the shortest delay.

To understand the impact of the distance to the closest landmark, we select all the probing hosts that have measured more than 100 landmarks. For each probing host, we find the distance between it and its closest landmark. Then, given a certain distance value d_i, we calculate the average d_r of probing hosts for which the distance to the closest landmark is within d_i, as shown in Fig. 6. For the probing hosts whose distance of the closest landmark is 0, the average value of d_r is 2.924 percent; and for the probing hosts whose distance of the closest landmark is within 500 km, the average value of d_r is 4.140 percent. It is clear that when the distance of the closest landmark increases, the value of d_r also increases. It is easy to explain the results. When the distance of the closest landmark is smaller, there is higher probability that the landmark with the shortest distance returns the shortest delay. We also find that there is a jump in the curve when the distance is 300 km. It is because that in China, most provinces are located 300 km far away from each other, and the intraprovince Internet connectivity is much richer than interprovince connectivity.

Therefore, it is very likely that closest-shortest rule will apply if probing sufficient number of landmarks, especially when the distance to the closest landmarks is small. Note that although our investigation is conducted on the data set from a moderately connected Internet region, the rule should also apply to richly connected Internet regions, since a linear delay-distance correlation implies closest-shortest rule.

3.4 Conclusions

We have studied the large data set collected from a moderately connected Internet region (China), and have the following observations:

- The linearity between delay and distance is positive but very weak.
- With high probability that closest-shortest rule holds if we probe a sufficient number of landmarks. The situation is even better when the distance to the closest landmark is small.

4 GeoGet: A Novel IP-Geolocation Mapping Scheme

In this section, we develop a simple and novel IP-geolocation mapping scheme called GeoGet. In practice, GeoGet can be deployed in combination with a certain locality-aware application and the application can thus collect the geolocations of their clients.

4.1 Design Goals

GeoGet is designed specifically for moderately connected Internet regions, and it has the following design goals:

1. Mapping an IP address to a city-level geolocation with small error distance.
2. No need to install client-side software for delay measurement.
3. Controlling the measurement cost for a targeted client.

In what follows, we present the detailed techniques in GeoGet to meet the design goals.

4.2 Using Webservers as Passive Landmarks (LM)

Based on the analysis in the previous section, the closest-shortest rule is more applicable than delay-distance correlation for moderately-connected Internet regions. Therefore, in GeoGet, we map a targeted client to the same city as the landmark which has the shortest delay.

To cover targeted clients from diverse geolocations, GeoGet requires landmarks in all possible cities. In addition, as shown in Section 3, if we have more landmarks to probe, the closest-shortest rule holds better, and thus the mapping result will be more accurate. For this reason, it is desirable that we have multiple landmarks in a city. More landmarks will bring additional advantages too. First, the measurement load can be shared among landmarks; Second, the single-point failure can be avoided.

However, it is very difficult to actively deploy such a large number of landmarks with wide coverage. Our solution in GeoGet is to use webservers as passive landmarks. Given the popularity of web applications, there are a large number of webservers and their geolocations cover almost every city. Using webservers as passive landmarks totally eliminates the deployment and maintenance costs for active landmarks.

4.3 HTTP/Get Probing Using JavaScript at Targeted Clients

Since we use webservers as passive landmarks, the delay probing needs to be initiated from the client side. To avoid installing any client-side software, we use JavaScript to generate HTTP/Get probing at the targeted clients to measure the delays to the selected webserver landmarks. The JavaScript is stored at a webserver that a locality-aware application employs. When a client uses this service, it will automatically download and execute the JavaScript. The only requirement for the clients is that they have web browser installed and the browser supports JavaScript. The requirement can be easily met by all the desktop and laptop computers to date.

When executing the JavaScript code, the targeted client visits a nonexisting image in a certain webserver by HTTP/Get request and records the delay. The HTTP/Get request is sent multiple times and the minimum delay is assumed as the measured delay to the webserver. To bypass the possible web caches, each time the targeted client request for different nonexisting images.

We should make sure that networking delay is the dominant part for the delay measured by HTTP/Get probing. In other words, the server processing delay for HTTP/Get request should be quite small compared with networking delay.

To verify this, we have compared HTTP/Get probing with ICMP probing, by measuring the delays to the same set of webservers. Each webserver was probed 10 times, and the minimum value was chosen as the measured delay. We totally measured 8,000 webservers. Almost all the webservers responded to the HTTP/Get probing, but only 2,876 webservers responded to ICMP probings. It validates our arguments that ICMP probing is prohibited in many routers and servers. Fig. 7 shows the (ICMP, HTTP/Get) delay pairs, each for a webserver. We find that most (ICMP, HTTP/Get) pairs are near the ideal $y = x$ line. For more than 90 percent pairs, the difference between the ICMP delay and the HTTP/Get delay is within 10 percent. Considering networking jitters, we can see that there is no obvious difference between the delays of ICMP probing and HTTP/Get probing. Therefore, the server processing delay for HTTP/Get request is very small and can be omitted and networking delay is the dominant part in HTTP/Get probing. As a result, it makes sense to choose HTTP/Get as the probing tool in GeoGet.

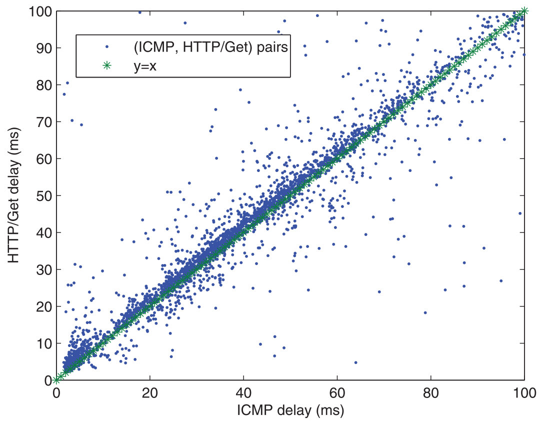


Fig. 7. Delay comparison between ICMP probing and HTTP/Get probing. Most (ICMP, HTTP/Get) pairs are near the ideal $y = x$ line.

4.4 Landmark Selection (LMS)

Given so many landmarks in GeoGet, the measurement cost is too high if a targeted client is to probe all landmarks. To control the measurement cost, it is desirable if we can select a subset of all the landmarks for a targeted client.

We adopt a two-step probing method to refine the geolocation of a targeted client. The first step is area-level probing, and the second step is city-level probing. All cities in the entire region are separated to a few number of *areas* according to their geolocations, and there is a *center city* in each area. In area-level probing, a number of landmarks from the center cities are selected for the targeted client. A controlled number of areas with shortest delays after area-level probing are chosen to enter city-level probing, in which the landmarks from each city of the chosen areas are selected. In this way, a targeted client does not need to probe landmarks from all cities.

We also need to select a number of landmarks from a single city for a targeted client. In GeoGet, we do not limit the candidate landmarks in a city to a few *powerful* landmarks, not only for concern of load balance, but also because that the webserver landmarks are not obligated to participate in GeoGet. Therefore, the measurement cost at landmark-side is divided into as many landmarks as possible, and the processing load put on a single landmark is relatively light.

As a whole, landmark selection for a targeted client needs to address the following issues. First, what is the number of landmarks to select from each city to probe. Second, after area-level probing, how many areas with shortest delays are selected to enter city-level probing. Third, how to select a certain number of landmarks from all landmarks in a city. For the former two issues, there is a tradeoff between mapping accuracy and probing cost, as implied by closest-shortest rule. For the third issue, we select the landmarks from the same Autonomous System (AS) as the targeted client with higher preference. This is because the delay between two hosts within the same AS is usually less than that from different ASes, and the shorter delay means better accuracy. We can also employ some network latency estimation approach such as GNP [30] or Vivaldi [31] to discover closer landmarks to the targeted client.

Algorithm 1 illustrates the pseudocode for landmark selection. $M1$ and $M2$ are the numbers of landmarks selected from a single city in area-level probing and city-level probing, respectively, and P is the number of areas to enter city-level probing. Since in most cases the targeted clients within one/24 IP segment are from the same city, we take them as identical. And note that a targeted client may visit the coordination server again after it completes probing partial landmarks.

Algorithm 1. Landmark selection algorithm
Input: T_IP (/24 IP prefix of the targeted client)
T_AS (AS number of T_IP)
// **Area-level landmark selection**
01 $CSET1 =$ center cities that T_IP has not probed
02 $LSET1 = \phi$
03 **for** each *city1* in *CSET1*
04 $LC1 =$ landmarks in *city1* within T_AS
05 $LC0 =$ landmarks in *city1* within ASes other than T_AS
06 **if** $|LC1| \geq M1$
07 $LC2 =$ randomly select *M1* landmarks from *LC1*
08 $LSET1 = LSET1 \bigcup LC2$
09 **else**
10 $LC2 =$ randomly $(M1 - |LC1|)$ landmarks from $LC0$
11 $LSET1 = LSET1 \bigcup LC1 \bigcup LC2$
12 **end if**
13 **end for**
14 Assign $LSET1$ for targeted client to probe
// **City-level landmark selection**
15 $ASET1 = P$ areas to enter city-level probing
16 $CSET2 =$ cities in *ASET1* that T_IP has not probed
17 $LSET2 = \phi$
18 **for** each *city2* in *CSET2*
19 $LC1 =$ landmarks in *city2* within T_AS
20 $LC0 =$ landmarks in *city2* within ASes other than T_AS
21 **if** $|LC1| \geq M2$
22 $LC2 =$ randomly select *M2* landmarks from *LC1*
23 $LSET2 = LSET2 \bigcup LC2$
24 **else**
25 $LC2 =$ randomly select $(M2 - |LC1|)$ landmarks from $LC0$
26 $LSET2 = LSET2 \bigcup LC1 \bigcup LC2$
27 **end if**
28 **end for**
29 Assign *LSET2* for targeted client to probe

Lines 1-14 describe the area-level landmark selection. Assume the/24 IP prefix of a coming targeted client is T_IP and its AS number is T_AS. First check the center cities that T_IP has not probed. Then, for each of the unprobed center cities, if there are more than $M1$ landmarks within T_AS from the city, randomly select $M1$ such landmarks; otherwise, select all the landmarks within T_AS from the city and the remaining landmarks within other ASes from the city. The selected area-level landmarks are then assigned to targeted client to probe.

Lines 15-29 show the city-level landmark selection. First check the unprobed cities from the P areas entering city-level probing. Then, for each of the unprobed cities, select $M2$ landmarks from it, according to the same method in area-level landmark selection. Finally, assign the selected city-level landmarks to targeted client to probe.

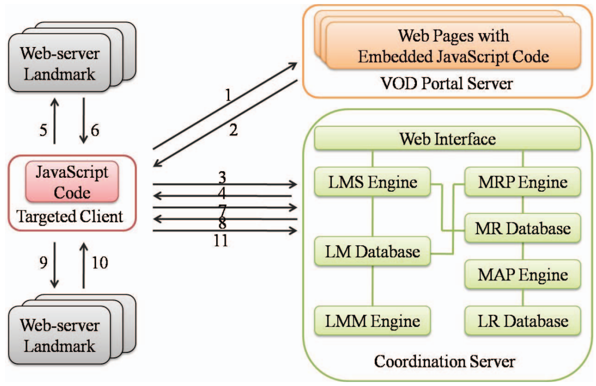


Fig. 8. Implementation architecture of GeoGet.

5 Implementation

We have implemented GeoGet by using around 40,000 webservers with known geolocations as passive landmarks, and deploying a coordination server in combination of a website providing video-on-demand service. In practice, multiple servers can be packed together to play as the coordination server, in order to avoid the system bottleneck. During the running time we measured, GeoGet attracts more than 5,000 clients from diverse geolocations to visit and participate in the measurement study by the time when writing the paper. In this section, we describe the implementation details.

5.1 Architecture

Fig. 8 illustrates the implementation architecture, which consists of four parts: the video-on-demand portal server, the coordination server, the passive landmarks, and the targeted clients. The VOD portal server is deployed for this particular study to attract clients to participate in the measurement study. We embed some JavaScript code in the webpages of the portal server. When a client visits the portal server, it gets the webpages and executes the JavaScript code embedded. The client interacts with the coordination server to carry out delay measurement during the time period when the client streams video clips from the portal server. The landmarks are just regular webservers and only need to passively respond to the HTTP/Get probings from the client. All the measured results are reported to the coordinate server which finally maps each client to a city.

5.1.1 Webserver Landmarks

To collect webserver landmarks, we have crawled a huge number of webservers in China and got their IP addresses. Then, we check the geolocations of the webservers by multiple IP-geolocation mapping databases. Only those webservers whose geolocations are agreed by all the databases are chosen as passive landmarks in our system. Therefore, the error rate of the geolocations of the webserver landmarks is neglectable.

Finally, we get 43,973 webservers as passive landmarks. The webservers cover 336 cities, out of a total number of 346 cities in China. The coverage ratio is 97.1 percent. Considering that the Internet popularity in China is uneven, there may be a few clients coming from cities where we cannot find any webserver. But overall, the coverage ratio is adequate for most locality-aware applications.

5.1.2 Coordination Server

The coordination server is the core part of our system as it guides the targeted clients to perform all the measurement tasks, and process the measurement results. It consists of seven main components: three databases as well as four engines.

Landmark database. LM Database stores the information of the webserver landmarks we use, including their IP addresses, geolocations as well as some status information, such as TIMEOUT errors reported by clients.

Measurement result database. MR Database stores both the area-level and city-level measurement results, including the IP addresses of the targeted clients and the corresponding landmarks probed, as well as the measured delays. All the measurement results are indexed using the corresponding/24 IP segment so that the measurement progress of a given/24 IP segment can be quickly checked.

Location result (LR) database. LR Database stores the geographical mapping results of targeted clients, including the IP addresses of targeted clients and the corresponding mapping cities.

Landmark selection engine. LMS Engine implements our landmark selection algorithm described in Algorithm 1. Given a targeted client, it first queries LM Database to get a set of landmarks based on our landmark selection algorithm, then queries MR Database to check the measurement progress of the targeted client, and finally sends the unprobed landmarks to the targeted client for delay measurement.

Landmark maintenance (LMM) engine. The conditions of webservers keep changing, e.g., some webservers change their configurations and some webservers will even die. We need to track the status of our landmarks. LMM Engine is a background engine used for dynamic maintenance on LM Database, which includes cleaning out the landmarks for which many TIMEOUT errors are reported, as well as adding new webservers as passive landmarks.

Measurement result processing (MRP) engine. MRP Engine is responsible for processing the measurement results from targeted clients. It updates the MR Database using the received measurement delays and if a TIMEOUT error is reported, it also updates the LM Database to record the event with the corresponding landmark reported by the targeted client.

MAP (IP-geolocation mapping) engine. If a targeted client finishes the city-level probing, MAP Engine checks the corresponding landmark information and delay information stored in MR database, uses the closest-shortest rule to map the client to a city, and adds the mapping result to LR Database.

5.2 Probing Cost

We estimate the probing cost for a targeted client in terms of the number of landmarks, based on the parameters set in our system.

The number of landmarks probed depends on two factors: 1) how many cities to probe; and 2) how many landmarks to probe for each city. If there are totally C cities and they are separated to A areas, $M1$ landmarks are selected from the center city of each area in area-level probing, P areas with shortest delays after area-level probing are chosen to enter city-level probing, and $M2$ landmarks are selected from each city in city-level probing, the expected number of landmarks probed for a targeted client is $M1 * A + M2 * P * C/A$.

The passive landmarks we collect cover 336 cities in China, i.e., $C = 336$. In our implementation, we set $A = 31$, $M1 = M2 = 2$, $P = 2$, so the average number of landmarks probed is 102. But of course if we can bear less mapping accuracy and set $M1 = M2 = 1$ and $P = 1$, the number of landmarks probed becomes 51 and the probing cost is reduced by 50 percent. When there are multiple targeted clients from one/24 IP segment, the average number of landmarks probed for each client is even less than the expected value, since the clients will cooperatively complete the delay measurement.

6 Evaluation

There are more than 5,000 clients which have visited our coordination server. To estimate the mapping accuracy, we should know the actual geolocation of the clients. Among them, we further select the clients that have finished measuring the landmarks as we require for evaluation. It is challenging to get the actual ground truth for the clients and webservers. For the webserver landmarks, their geolocations are obtained from Structon [21]. Structon takes a web mining approach to infer the geolocations of webservers, and its city-level accuracy is 87.4 percent, which is the database with the highest mapping accuracy we can get. As for the ground truths of the clients, we use three IP-Geolocation databases, namely, IP138 [10], Quova [9], and IP2location [11]. IP138 is an IP-Geolocation database which collects IP addresses of China. Quova and IP2location are both commercial databases. The mapping accuracy of these databases are claimed over 95 percent in state/province level [29]. Only the clients whose physical locations are agreed by all the three databases are selected in the final statistics. Hence, there is no/little bias on the measurement results. There are about 2,000 such clients. We also add in

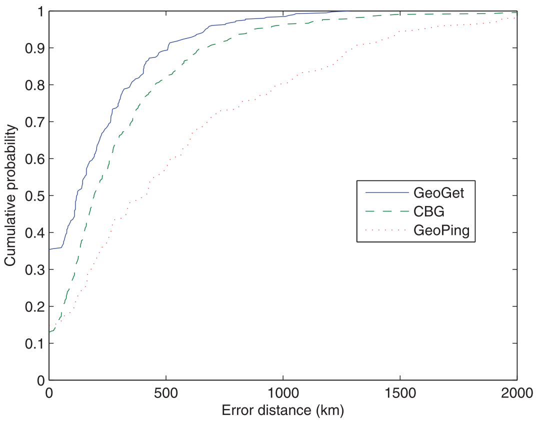


Fig. 9. CDF of error distances of GeoGet, GeoLim, and GeoPing.

40 PlanetLab nodes (from some cities) as the clients. The evaluations below are conducted on all the clients.

We still use C to denote the total number of cities covered by our webserver landmarks, A to denote the number of areas, $M1$ and $M2$ to denote the number of landmarks selected from a city in area-level probing and city-level probing, respectively, and P to denote the number of areas to enter city-level probing.

6.1 GeoGet versus GeoLim and GeoPing

We compare the mapping accuracy of GeoGet with GeoLim [16] and GeoPing [14]. We do not compare with TBG [17] or Octant [18] because the mapping method of TBG is similar to GeoLim except that TBG takes use of many intermediate routers as landmarks, and Octant assumes there is a nonzero lower bound of distance/delay value, which we did not find from our data set (refer to Section 3).

To make fair comparison, we use the same landmarks for the three schemes we evaluate. The parameters in GeoGet is set as $C = 336$, $A = 31$, $M1 = 2$, $P = 2$, and $M2 = 2$. Therefore, the total number of landmarks probed by a targeted client is about 102. For GeoLim, the landmarks assigned to a targeted client are exactly the same as those measured in GeoGet, including both the area-level landmarks from center cities and the city-level landmarks from the cities of the two areas entering city-level probing. For GeoPing, we choose 20 probing hosts and also select the landmarks measured in GeoGet as the landmarks.

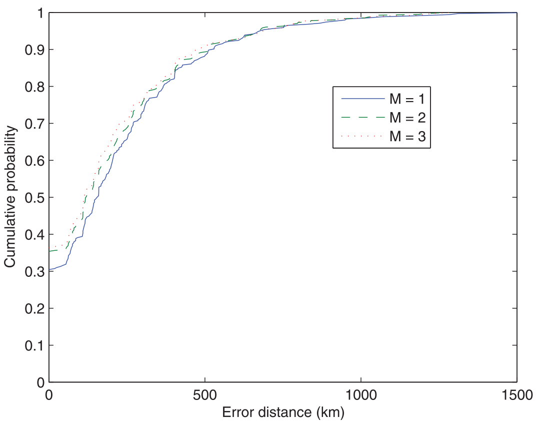


Fig. 10. CDF of error distances if selecting M landmarks from a city.

TABLE 1
Selecting M Landmarks from a City

M	1	2	3
City-level accurate ratio	30.4%	35.4%	36.3%
Median error distance in terms of city (km)	160	120	110
Average error distance in terms of city (km)	220	200	190
Number of landmarks probed	51	102	153

GeoLim fails to form an intersection region for 29 targeted clients. The CDF of error distances of the three schemes are shown in Fig. 9. In GeoGet, we can accurately map 35.4 percent targeted clients to city level, outperforming GeoLim and GeoPing by 270 and 239 percent, respectively. The median error distance in terms of city in GeoGet is around 120 km, outperforming GeoLim and GeoPing by about 37 and 70 percent, respectively. And the average error distance in terms of city in GeoGet is around 200 km, outperforming GeoLim and GeoPing by about 34 and 65 percent, respectively.

It is easy to explain the comparison result. Both GeoLim and GeoPing are based on strong delay-distance correlation. In GeoLim, the distance is extracted from the measured delay value. In GeoPing, the shortest euclidean distance in distance dimension is mapped to that in delay dimension. But the delay-distance correlation is weak in the moderately connected Internet region we study. Therefore, the evaluation results inversely verify the supposition we make in Section 3, that is, the closest-shortest rule is more applicable than delay-distance correlation if including moderately-connected Internet regions.

6.2 Number of Landmarks Selected from a City

As we investigate in Section 3, the closest-shortest rule holds with higher probability if measuring more landmarks. In GeoGet, there are more than one passive landmarks in a single city. Selecting more landmarks from a city (here, we simply assume that the numbers of landmarks selected from a city in area-level probing and city-level probing are the same, that is, $M1 = M2 = M$) will add the probing cost, but at the same time it can bring higher mapping accuracy.

To study the tradeoff between mapping accuracy and the number of landmarks selected in a city, M, we vary M as 1, 2, and 3, respectively. The other parameters are set as $C = 336$, $A = 31$, $P = 2$. The resultant CDF of error distances are shown in Fig. 10. Table 1 also illustrates the city-level accurate ratio, median error distances, and average error distances in terms of city, as well as the correspondent number of landmarks probed for each case. We find that the mapping accuracy becomes better if selecting more landmarks from a city. Though the mapping accuracy of GeoGet is not very high, we believe the accuracy will improve significantly if probing much more landmarks from each city.

6.3 Number of Areas to Enter City-Level Probing

We adopt a two-step probing method to refine the geolocation of a targeted client, first area-level probing and then city-level probing. After area-level probing, if we choose more areas with shortest delays to enter city-level

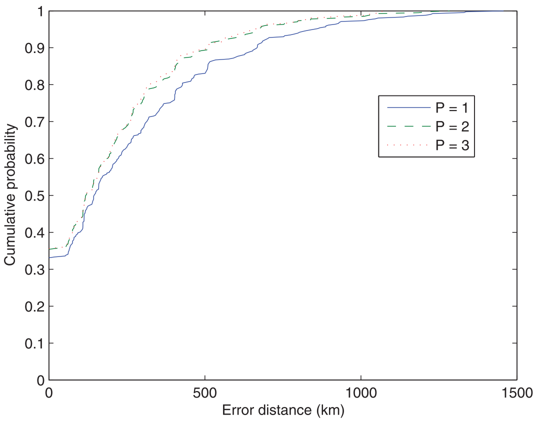


Fig. 11. CDF of error distances if selecting P areas to enter city-level probing.

probing, the mapping accuracy will also improve. In other words, there is also a tradeoff between the mapping accuracy and the number of areas to enter city-level probing.

Assume the number of areas chosen to enter city-level probing is P. We vary P as 1, 2, and 3, respectively, and the other parameters are set as $C = 336$, $A = 31$, $M1 = M2 = 2$. The CDF of error distances are shown in Fig. 11. Table 2 gives numerical illustrations. If increasing the number of P, the error distance indeed decreases. But the decrease from $P = 2$ to $P = 3$ is quite marginal. Considering the number of landmarks probed, we set $P = 2$ as the number of areas to enter city-level probing in our system.

7 Conclusion

In this paper, we investigate the delay-distance relationship in China, which is the world's largest country in the number of Internet users and the second largest in the size of IP address space. We find that the linearity between delay and distance is positive but very weak. However, the closest-shortest rule holds with high probability. For IP-Geolocation mapping in moderately connected Internet regions, we develop GeoGet. GeoGet adopts closest-shortest rule as the mapping principle, and does not depend on delay-distance correlation as prior work. GeoGet takes use of a large number of webservers as passive landmarks. JavaScript code is embedded in webpages of locality-aware applications for clients to execute when visiting the site. The delay measurement can thus be carried on at targeted clients using HTTP/Get probing generated by JavaScript, without any client-side software installation. Further, we adopt a two-step probing method to refine the geolocation of a targeted client, first to area-level and then to city-level. We have implemented GeoGet, and the evaluation results shows that the mapping accuracy of GeoGet significantly outperforms traditional IP-Geolocation schemes such as GeoLim and GeoPing.

Acknowledgments

This work is supported by the National Basic Research Program of China (973 Program) under Grant 2011CB302900 and 2009CB320501, the National Natural Science Foundation of China (Nos. 61170291, 61133006, 61120106008).

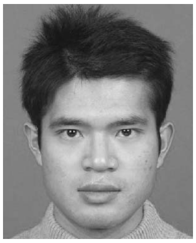
TABLE 2
Selecting P Areas to Enter City-Level Probing

P	1	2	3
City-level accurate ratio	33.1%	35.4%	35.5%
Median error distance in terms of city (km)	150	120	120
Average error distance in terms of city (km)	240	200	190
Number of landmarks probed	82	102	122

References

[1] H. Xie et al., "P4P: Provider Portal for (P2P) Applications," *Proc. ACM SIGCOMM '08*, 2008.
[2] V. Aggarwal, A. Feldmann, and C. Scheideler, "Can ISPs and P2P Users Cooperate for Improved Performance?" *ACM SIGCOMM Computer Comm. Rev.*, vol. 37, no. 3, pp. 29-40, 2007.
[3] R. Bindal et al., "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '06)*, 2006.
[4] K. Xu et al., "LBMP: A Logarithm-Barrier-Based Multipath Protocol for Internet Traffic Management," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 3, pp. 476-488, Mar. 2011.
[5] China Internet Network Informationi Center, http://www.cnnic.net.cn/, 2012.
[6] Whois.net, http://www.whois.net/, 2012.
[7] Univ. of Illinois at Urbana-Champaign: (IP Address to Latitude/Longitude), http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll/, 2011.
[8] D. Moore et al. "Where in the World Is Netgeo.Caida.Org?," *Proc. Int'l Networking Conf. (INET '00)*, 2000.
[9] Quova, http://www.quova.com, 2012.
[10] IP138, www.ip138.com, 2012.
[11] IP Address Geolocation to Identify Website Visitor's Geographysical Location, http://www.ip2location.com/. 2012.
[12] IPInfoDb, http://ipinfodb.com/, 2012.
[13] IANA-Internet Assigned Numbers Authority, http://www.iana.org/, 2012.
[14] V. Padmanabhan and L. Subramanian, "An Investigation of Geographic Mapping Techniques for Internet Hosts," *Proc. ACM SIGCOMM '01*, 2001.
[15] A. Ziviani et al. "Improving the Accuracy of Measurement-Based Geographic Location of Internet Hosts," *Computer Networks*, vol. 47, no. 4, pp. 503-523, 2005.
[16] B. Gueye et al., "Constraint-Based Geolocation of Internet Hosts," *Proc. ACM Internet Measurement Conf. (IMC '04)*, 2004.
[17] E. Katz-Bassett et al. "Towards IP Geolocation Using Delay and Topology Measurements," *Proc. ACM Internet Measurement Conf. (IMC '06)*, 2006.
[18] B. Wong, I. Stoyanov, and E. Sirer, "Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts," *Proc. USENIX Conf. Networked Systems Design and Implementation (NSDI '07)*, 2007.
[19] T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations," *Survey Rev.*, vol. 22, no. 176, pp. 88-93, 1975.
[20] B. Eriksson et al., "A Learning-Based Approach for IP Geolocation," *Proc. Int'l Conf. Passive and Active Measurement (PAM '10)*, 2010.
[21] C. Guo et al., "Mining the Web and the Internet for Accurate IP Address Geolocations," *Proc. IEEE INFOCOM '09*, 2009.
[22] I. Poese et al., "IP Geolocation Databases: Unreliable?" *ACM Computer Comm. Rev.*, vol. 41, no. 2, pp. 53-56, Apr. 2011.
[23] J. Muir and P. Oorschot, "Internet Geolocation: Evasion and Counterevasion," *ACM Computing Surveys*, vol. 42, no. 1, 2009.
[24] S. Laki, P. Matray, and P. Haga, "Spotter: A Model Based Active Geolocation Service," *Proc. IEEE INFOCOM '11*, 2011.
[25] M. Arif, S. Karunasekera, and S. Kulkarni, "GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements," *Proc. 33rd Australasian Conf. Computer Science (ACSC '10)*, 2010.
[26] B. Gueye, S. Uhlig, and S. Fdida, "Investigating the Imprecision of IP Block-Based Geolocation," *Proc. Int'l Conf. Passive and Active Network Measurement (PAM '07)*, 2007.

[27] B. Gueye et al., "Leveraging Buffering Delay Estimation for Geolocation of Internet Hosts," *Proc. Int'l IFIP-TC6 Conf. Networking Technologies, Services, and Protocols (Networking '06)*, 2006.
[28] Y. Wang et al., "Towards Street-Level Client-Independent IP Geolocation," *Proc. USENIX Conf. Networked Systems Design and Implementation (NSDI '11)*, 2011.
[29] Geolocation: Core To The Local Space And Key To Click-Fraud Detection, http://searchengineland.com/geolocation-core-to-the-local-space-and-key-to-click-fraud-detection-11922, 2012.
[30] T. Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-Based Approaches," *Proc. IEEE INFOCOM '02*, 2002.
[31] F. Dabek et al., "Vivaldi: A Decentralized Network Coordinate System," *Proc. ACM SIGCOMM '04*, 2004.



Dan Li received the BS degree in computer science from Beijing Normal University in 2003. He received the ME and PhD degrees from Tsinghua University in 2005 and 2007, respectively, both in computer science. He joined the Computer Science Department of Tsinghua University in March 2010, where he is now an associate professor. From January 2008 to February 2010, he worked as an associate researcher in the Wireless & Networking Group of Microsoft Research Asia. His current research interests include computer network architecture, data center networks and green networking. He has published more than 30 technical papers in referred conferences and journals, including top conferences (e.g., SIGCOMM, ICNP) and top journals (e.g., ToN, TMM) in his research fields. He serves as a TPC member for some international conferences, such as INFOCOM, GLOBECOM, ICCCN. He is a member of the IEEE and the ACM.

Jiong Chen is currently working toward the master's degree from Pecking University, China. His main research area is computer networks and measurement.

Chuanxiong Guo received the PhD degree in communications and information systems from Nanjing Institute of Communications Engineering, China, in 2000. He is currently a lead researcher in the Wireless and Networking Group at Microsoft Research Asia. His research interests lie in the field of networking, encompassing network algorithm design and analysis, data center networking (DCN), novel network applications, and networking support in operating systems.

Yunxin Liu received the double BS degrees in mechanical engineering and computer science from the University of Science and Technology of China, HeFei, Anhui Province, P.R. China, in 1998, the MS degree in computer science from Tsinghua University, Beijing, P.R. China, in 2001, and the PhD degree in computer science from Shanghai Jiao Tong University, P.R. China, in 2011. His research interests include Computer Networking, System Design and Mobile Computing. He is a member of the IEEE and the ACM.

Jinyu Zhang is currently an associate professor in Peking University. His main research interest includes computer networks, multimedia, as well as network measurement.

Zhili Zhang received the MS and PhD degrees in computer science from the University of Massachusetts in 1992 and 1997 (PhD thesis advisors, Profs. Don Towsley and Jim Kurose). He joined the Department of Computer Science and Engineering at the University of Minnesota in January 1997, where he is now a full professor. His research interests lie broadly in computer communication and networks, Internet technology, multimedia and emerging applications.

Yongguang Zhang received the PhD degree in computer science from Purdue University in 1994. He is a senior researcher and research manager for the Wireless and Networking research group at Microsoft Research Asia. Previously from 1994 to 2006, he was a senior research scientist at HRL Labs. He was also an adjunct assistant professor of computer science at the University of Texas at Austin from 2001 to 2003. He has published more than 50 technical papers and one book, including top conferences and journals of his fields (like Sigcomm, Mobicom, Mobisys, ToN). He recently won the Best Paper Award at NSDI '09, and four Best Demo Awards in a roll: at MobiSys '07, at SenSys '07, again at MobiSys '08, and at NSDI '09. He is an associate editor for *IEEE Transactions on Mobile Computing*, an area editor for MC2R, was a guest editor in an *ACM MONET Journal*, and has organized and chaired/co-chaired several international conferences, workshops, and an IETF working group. He was a general co-chair for ACM MobiCom '09.