Towards IP Geolocation Using Delay and Topology Measurements

Ethan Katz-Bassett*   John P. John*   Arvind Krishnamurthy*   David Wetherall†
Thomas Anderson*   Yatin Chawathe‡

ABSTRACT

We present Topology-based Geolocation (TBG), a novel approach to estimating the geographic location of arbitrary Internet hosts. We motivate our work by showing that 1) existing approaches, based on end-to-end delay measurements from a set of landmarks, fail to outperform much simpler techniques, and 2) the error of these approaches is strongly determined by the distance to the nearest landmark, even when triangulation is used to combine estimates from different landmarks. Our approach improves on these earlier techniques by leveraging network topology, along with measurements of network delay, to constrain host position. We convert topology and delay data into a set of constraints, then solve for router and host locations simultaneously. This approach improves the consistency of location estimates, reducing the error substantially for structured networks in our experiments on Abilene and Sprint. For networks with insufficient structural constraints, our techniques integrate external hints that are validated using measurements before being trusted. Together, these techniques lower the median estimation error for our university-based dataset to 67 km vs. 228 km for the best previous approach.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Algorithm, Measurement

Keywords

Geolocation, Network topology, Delay measurements, Route measurements.

---

1. INTRODUCTION

The ability to determine the geographic location of an Internet host would enable a variety of applications, from the mundane to the essential. Commercial databases currently provide rough and incomplete location information, enabling some targeted advertising delivered by the Web, as well as other content localization. If dependable, it could serve as part of an E-911 system for voice-over-IP or a broadcast system for regional emergencies. A ubiquitous location service as part of the infrastructure has been identified by some as an important vision for the future Internet [5, 13].

We refer to the process of finding the geographic location of an Internet host as *IP geolocation*. This is a difficult problem, even putting mobility aside, because the decentralized management of the Internet means that there is no authoritative database of host locations. The databases that do exist are derived by combining a mish-mash of sources (including DNS LOC records, whois registration, and DNS hostname parsing rules). They are all manually maintained, and thus subject to inconsistencies and outdated information. Automated systems are desirable as they can eliminate these problems and produce dependable results. However, they exist only for specialized cases and equipment, such as the use of GPS [8] and GSM or 802.11 beacons [13]; even the latter depend on a large database of landmarks that must be manually entered.

Our larger goal is to develop an automated service for IP geolocation that is broadly applicable and scales to the size of the Internet. Such a service would be queried by IP address and return an accurate location estimate. It would have key advantages relative to existing systems. Unlike GPS, GSM and 802.11 methods, it would require no specialized hardware and thus be truly ubiquitous, available for all Internet hosts. Unlike methods based on DNS names [15, 18], it would automatically derive the location estimate even if DNS names are unavailable or incorrect, a common occurrence in high-churn databases.

In this paper, we consider the core problem that must be solved to enable such a service: how to estimate the location of a host given its IP address. To devise an automated solution, we focus on the use of network measurements. Since we are not the first to do so, we began our research by studying techniques proposed by others. These techniques are based on end-to-end delay measurements from a set of landmarks with known locations [18, 10]. As we experimented with variations and evaluated them using a dataset gathered on PlanetLab, we were surprised to discover that much simpler delay-based algorithms were able to deliver performance that was as good as or better than the state-of-the-art.

We further found that the error of all the pure delay algorithms we studied was strongly determined by the distance to the nearest landmark. This effect is due to the circuitousness and irregularity of Internet paths, and techniques that combine delays across land-

marks do little to help. The consequence is that delay-based algorithms must use many landmarks that are carefully chosen for their coverage if they are to consistently perform to any reasonable level of accuracy. Because of the difficulty of finding landmarks uniformly everywhere, these algorithms will typically work poorly for a fraction of targets; there are estimates that are more than 1000 km off in our US-based experiments.

This line of reasoning led us to conclude that other kinds of techniques were needed. To this end, we investigate algorithms that combine delay with topology to factor out the circuitousness of Internet paths. Inspired by algorithms used for localization in sensor networks [7, 4], we convert Internet route measurements into a set of constraints on the unknown locations of targets and intermediate routers en route to it, and then simultaneously geolocate the target and all of the routers in a way that best satisfies the constraints. This approach, which we refer to as Topology-based Geolocation (TBG), has a number of desirable properties: it takes advantage of the fact that routers nearby landmarks are easy to locate; it uses the locations of intermediate routers to quantify the directness of paths to targets, thus making these measurements more useful; and it allows the solution to be iterated using the coupling between network elements until all the elements have converged to an overall map that is consistent, as it must be in reality. TBG reduces the average error on targets on the structured Abilene and Sprint networks by about 40% and 70%, respectively, and the 90th percentile of error by a factor of four.

However, our study reveals that techniques based solely on network measurements have inherent limitations. For instance, if the Internet routes to the target do not sufficiently constrain the target's position – as when the tail ends of all routes converge to a shared segment that is of significant latency – then it is not possible to accurately geolocate the target without other sources of information. Fortunately, our topology-based technique can validate and incorporate locations of "passive" reference nodes – nodes that cannot issue measurements, but can be probed by "active" landmarks – to help constrain the topology. It generates the network topology containing the target and the passive reference nodes, uses delay and topology measurements to validate the positions of the passive nodes, and then derives location estimates for the target based on the entire topology. We improve the median error for difficult targets by more than a factor of three compared to established techniques. We believe the approach shows promise as a direction for future IP geolocation work.

The rest of this paper is organized as follows. In Section 2, we present the problem in more detail and review related work. Section 3 presents and evaluates new and existing variations on delay-based geolocation techniques, and identifies some of their limitations. Section 4 then presents a geolocation technique that also takes into account the structure of the Internet and its routing, and Section 5 evaluates its performance compared to that of delay-based techniques. We conclude by discussing the strengths and weaknesses of the different techniques.

2. PROBLEM AND PRIOR WORK

2.1 Problem

The version of the IP geolocation problem we consider in this paper is how to automatically estimate the coarse-grained geographic location of an arbitrary computer on the Internet. By automatic, we mean that the technique should not rely on human input other than to establish geographic coordinates for reference hosts; all schemes require some ground truth to bootstrap the system, but a small set of reference hosts should enable the location of a much larger set of targets. Furthermore, probable locations of nodes, if provided by outside sources, must be validated automatically by the reference hosts before they can be used in geolocating targets. By arbitrary computer, we mean that the technique should be able to locate all IP addresses, rather than a subset that belong to a particular provider, have registered in some manner, and so forth. By coarse-grained location, we mean that estimates should be accurate to within about the level of a major metropolitan area. Tighter estimates are desirable, but city-level estimates would already be sufficient for many applications.

In this paper, we consider IP geolocation techniques based on network measurements. In this setting, we have a set of reference hosts with known locations that we refer to as *landmarks*. Some are *active landmarks* that can issue probes, while some may be *passive landmarks* that cannot. Elsewhere in this paper, we will specify that landmarks are passive when the distinction is important; otherwise, they can be assumed to be active. We gather measurements between the landmarks and other hosts with unknown locations called *targets*, as well as between the landmarks. We then process the measurements according to a specified algorithm to estimate the locations of the targets. Because we want to be able to map hosts without having to first upgrade their software, we only consider algorithms that can be run using measurements that originate at landmarks, e.g., measuring the path and delay to targets. We do not use any measurements that originate at targets.

2.2 Internet Measurement Techniques

Two published geolocation techniques fit our problem and approach, GeoPing [18] and Constraint-Based Geolocation (CBG) [10]. Both use delay measurements from landmarks to estimate the location of Internet hosts. We present them in some detail below, because they show how delays may relate to location in non-trivial ways, and because we will build on them shortly (Section 3).

2.2.1 GeoPing

GeoPing locates a target by mapping it to the most representative landmark and using the location of the landmark as the estimate for the location of the target [18]. To do so, GeoPing assumes that two hosts that are near each other will measure similar delays to other landmarks. A target is probed from all landmarks to build a delay vector that acts as a profile of how "near" the landmarks are. The target is mapped to the landmark with the most similar profile. Similarity is computed as the Euclidean distance between delay vectors, the distance to the target in "delay space."

Interestingly, GeoPing can augment its set of landmarks with a set of passive ones that cannot perform probes to the target. In such settings, the target can be mapped either to an active or a passive landmark. This setting is interesting because it is "cheaper" to add passive landmarks, and they might allow techniques to perform better without increasing the density of probing landmarks.

2.2.2 Constraint-Based Geolocation

Instead of mapping targets to the location of a landmark, Constraint-Based Geolocation (CBG) employs a triangulation-like technique to combine delays from multiple landmarks and can return positions that lie between landmarks. To relate delay to distance, each landmark measures the delay from itself to all other landmarks. It then fits what is termed a *bestline* to this data. This is essentially the tightest line fit above all the *(delay, distance)* pairs. Figure 1 shows an example of the bestline for a Princeton University landmark. Because the bestline lies above all measured points, it converts delays into distance estimates that are taken to be upper bounds. The target is then assumed to be within a circle, centered at the landmark,
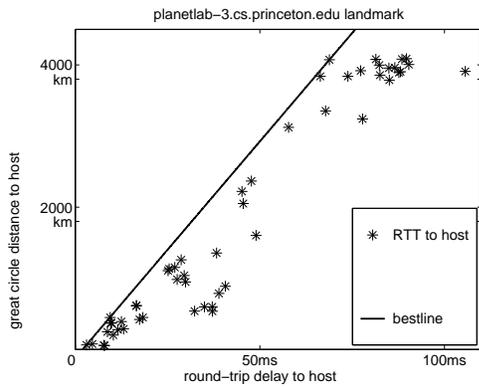
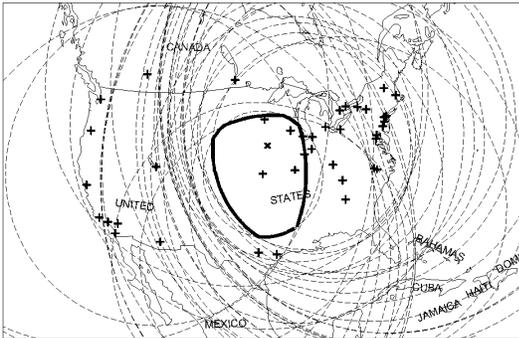**Figure 1: Example scatterplot and CBG bestline**

Figure 2: Example of constraint intersection region

whose radius is the estimated distance.

CBG then combines the distance estimates from all landmarks by intersecting all the circles. This intersection produces a feasible region in which the target is assumed to lie. The target is arbitrarily estimated to be located at the centroid of the region, and the size of the region is taken as a measure of the uncertainty (or confidence) in the estimate. Figure 2 shows an example. The '+' signs are landmarks, the dashed circles are constraints, and the intersection region is bounded by a bold perimeter. Experiments have shown CBG to provide better geolocation estimates than GeoPing and DNS-based approaches (e.g. [28]) on both United States and European datasets [10].

2.3 Other IP Geolocation Techniques

There are several geolocation techniques that can be used with Internet hosts but which we do not consider viable solutions to our problem.

The Global Positioning System (GPS) [8] provides accurate geographic locations for all hosts fitted with a GPS receiver. Over time, this hardware might be bundled with all computers. However, putting aside the obvious deployment challenge, GPS does not function well indoors or in urban canyons, limiting its ubiquity, and it would preclude many applications because it is client-driven.

Other systems such as Place Lab [13], Cricket [20] and RADAR [1] locate mobile hosts using 802.11 and GSM beacons with known locations. These systems have the potential for accurate estimates but their coverage is limited by the propagation range of APs and cell towers. Large-scale coverage requires wide-spread and dense deployment of nodes with 802.11 or GSM hardware, at known lo-

cations.

Systems such as IP2GEO [26], GeoCluster [18], GeoTrack [17], and Netgeo [27] require no special hardware. However, all these methods depend largely on manually maintained databases and are prone to incomplete coverage, outdated information, and faulty data entry. For example, Netgeo relies on DNS LOC records, whois registration records, and specialized DNS hostname rule files. DNS LOC records map IP addresses to locations but are rarely provided. Whois maps IP addresses to a registered administrative location, which may not reflect their actual location. Hostnames, especially for routers, may follow naming conventions that include geographic hints in the form of cities or airport codes [23], but many do not or are misleading [29].

2.4 Sensor Network Localization

Our topology-based techniques are inspired by similar techniques from the sensor network community, originally meant for positioning sensor nodes using observed radio connectivity [7, 4]. Observations that a pair of nodes are within or not within radio range induce a set of constraints on the locations of the nodes. The problem is then to solve for locations of target nodes, given a set of reference nodes with known locations. This problem can be formulated as a semidefinite program [25, 7, 4], allowing the use of powerful solvers such as Sedumi [21]. Our problem setting, however, requires a richer set of constraints. Nearby nodes may not be connected, and backbone links may connect nodes across the country. End-to-end delay measurements on the Internet translate to global constraints, where the position of a target is constrained by other nodes that are not nearby and the sum of the link distances should explain the end-to-end delay.

3. DELAY-BASED GEOLOCATION

In this section we present new techniques for delay-based geolocation. We find that simpler techniques can provide accuracy equivalent to GeoPing and CBG, but that all the delay-based techniques sometimes incur errors of 1000 km or more.

3.1 Dataset and Methodology

We used PlanetLab as a measurement platform to obtain the dataset that we use for experiments in this and other sections. We used 68 nodes at different PlanetLab sites in North America as our landmarks; these included all PlanetLab sites that had active nodes during our evaluation period. For the target set, we begin by evaluating existing techniques on a dataset comprising 128 hosts at universities across the US, since we could readily locate them. The targets were drawn from 37 out of 48 states on the mainland. We refer to this target dataset at the "University dataset."[1] Figure 3 shows a map of the landmarks and targets.

We envision geolocation on a global scale working in a hierarchical fashion: a small number of geographically dispersed landmarks first determines a general region (a continent, roughly) for the target location, then a denser set of landmarks in that region performs the final geolocation. Our PlanetLab dataset represents an example of one such dense set. Such a hierarchical organization reduces the number of probes necessary. The rough continent-level placement might even be done without measurement-based geolocation, through the use of DNS names or the mapping of IP addresses to countries through the Internet registries.

For techniques that can make use of passive landmarks, we use the same set of targets as passive landmarks. More specifically,

---

[1]The data for this dataset is available for download at http://pcp.cs.washington.edu/geolocation-data.tar
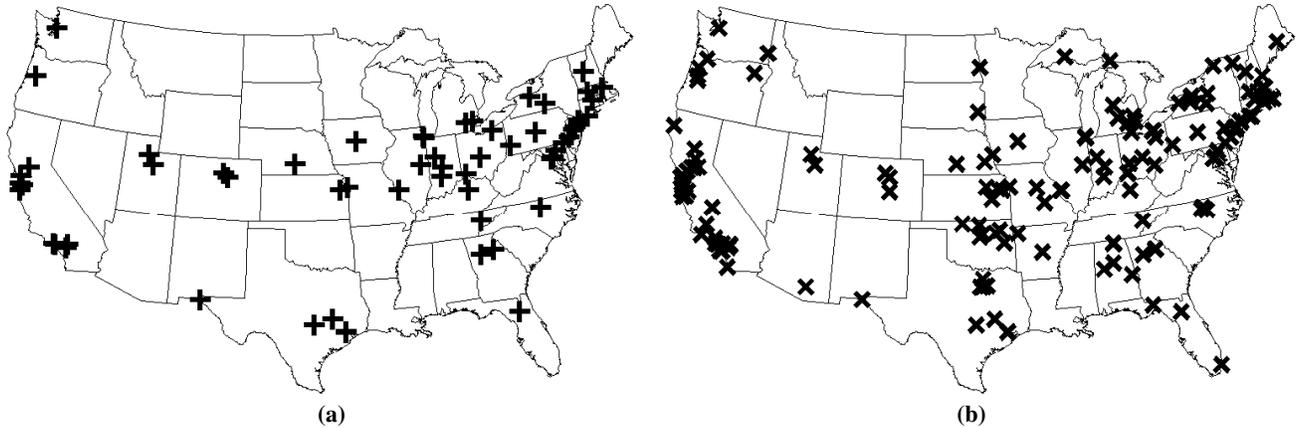
Figure 3: (a) Landmarks ('+'s) and (b) targets ('X's) in our University dataset.

while geolocating one of the targets, we assume that the locations of the remaining targets are known and the geolocation technique can use this information to better its accuracy.

This dataset provides a relatively large and geographically diverse dataset for which we have ground truth information[2] and from which we can make measurements. There are well-known network diversity issues associated with the use of PlanetLab and educational institutions [2]. We expect experiments with this dataset to show delay-based methods in a good light, and hence be appropriate for understanding the extent of their accuracy in controlled settings. This is because other datasets with richer inter-domain routing diversity exacerbate the difficulty of geolocation rather than make the problem simpler. After having established the limitations of the delay-based techniques even in these homogeneous settings, in Section 5.3, we evaluate the effectiveness of our topology-based algorithm on a set of non-academic targets.

We also believe that our comparison with GeoPing and CBG on this dataset is reasonable, because both techniques have been previously evaluated using measurements with a heavy academic bias. GeoPing was evaluated with university-based landmarks and targets [18], and CBG was evaluated with NLANR and PlanetLab data [10]. We expect these algorithms to be equally applicable to our dataset, and indeed judge them to perform on it comparatively as well as in their published evaluations.

To evaluate each technique, we compare the estimated locations for all targets to the corresponding ground truth locations. We generally look at the distribution of location errors since we care about consistent accuracy.

3.2 Shortest Ping

Shortest Ping is perhaps the simplest delay-based technique possible: each target is mapped to the landmark that is closest to it in terms of measured-round-trip time. Initially, we investigated it because we hoped to assess the value of more complex techniques by measuring their increase in accuracy. However, as we will illustrate in Section 3.5, we found it to provide results that are comparable with more complex algorithms. As with the other techniques that we discuss in this section, Shortest Ping requires the issuance of delay probes from each of the landmarks to the target. If the amount of probing traffic is an issue, one could use network coor-

dinates (such as Vivaldi [6]) and/or network location services (such as Meridian [3]) to first identify a small number of nearby landmarks and then issue probes only from the nearby landmarks to the target.

3.3 Speed of Internet (SOI) Geolocation

We use the term *constrained multilateration* (CM) to refer collectively to CBG and other delay-based techniques that combine distance constraints from multiple landmarks to arrive at a final estimate. CBG derives a distance-to-delay conversion on a per-landmark basis, based on measurements between landmarks. To evaluate the effectiveness of CBG's conversion, we give a simple variation, Speed of Internet (SOI), that uses a single conversion factor across all landmarks.
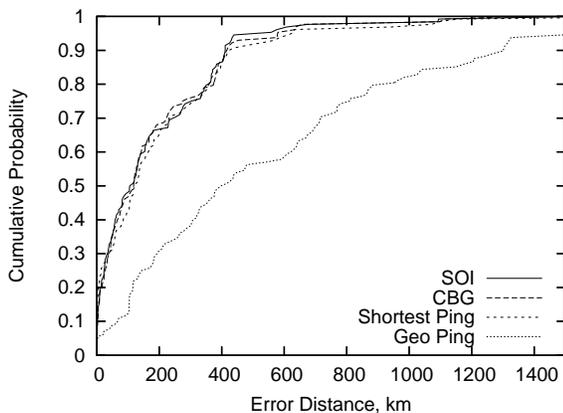
Data travels through fiber optic cables at almost exactly 2/3 the speed of light in a vacuum [19]. SOI is based on the observation that the geographic distance between hosts on the Internet is typically much less than the limit due to speed of light propagation, because circuitous paths, packetization, and other non-propagation delays can only inflate the time and reduce the effective rate of travel. This fact allows us to use constraints tighter than $\frac{2}{3}c$, where c is the speed of light in a vacuum. The intent of doing so is to narrow the intersection region without sacrificing location accuracy.

Nearly all measurements in our dataset exhibit end-to-end distances at most $\frac{4}{9}$ of what the speed of light allows given the delays. Over more than 25000 measurements from our PlanetLab landmarks, less than 0.04% of the measurements are for end-to-end distances that are more than $\frac{4}{9}$ of the speed of light limit. This makes it a safe threshold for constraints. More than 20% of the measurements exhibit end-to-end speeds that are at least $\frac{2}{9}$ of the speed of light, so the $\frac{4}{9}$ ratio is not overly loose.

SOI geolocation generates constraints using $\frac{4}{9}c$ as a time-to-distance conversion factor and is otherwise identical to CBG geolocation. It is much simpler than CBG as it does not require measurements between landmarks or calibration.

3.4 Underestimated Constraints

Both CBG and SOI use constraints that are less than the speed of light. This runs the risk of underestimates. When an underestimate occurs, the constraints either intersect in a region that does not contain the true location, or the constraints fail to intersect. For the University dataset, CBG constraints fail to form an intersection region for 27 of the 128 targets; in this case, as originally presented, CBG cannot return an estimate. SOI constraints do not form a re-

---

[2]Ironically, we used the PlanetLab administrative databases to locate their hosts and were hampered by a number of significant errors! We subsequently verified all locations to weed out these errors with a combination of USGS data [11] and Google Maps [12].

Figure 4: CDFs of location errors for delay-based geolocation techniques.

gion for 3 of the 128 targets. When there is no intersection region, one cannot know which among the set of constraints are the underestimates in order to fix them, since the actual location of the target is unknown. To guarantee that the techniques provide estimates for all targets, in the case of no intersection region, we use the speed of light in fiber ($\frac{2}{3}c$) to generate constraints. We use this modification as part of both CBG and SOI.

Note that underestimates limit the effectiveness of CM techniques because even when the intersection region exists it may not contain the target host. For 4 of the 101 targets for which the constraints form an intersection region, CBG generates such a false region. SOI constraints yield a false region for 10 of the 125 targets for which they form a region.

3.5 Comparison of Delay-based Techniques

We evaluated the delay-based techniques discussed in this section using the University dataset. Figure 4 compares delay-based techniques. GeoPing is evaluated using passive landmarks – when we locate a target, we use the remaining targets as passive landmarks. We make the following observations regarding the experimental results.

SOI and CBG give roughly the same quality of estimates. We also observe the rather surprising phenomenon that the simplest delay-based technique, Shortest Ping, performs only marginally worse than SOI and significantly better than GeoPing for our dataset.

But, more significantly, all the techniques provide very poor estimates for many of the targets, with worst-case errors of over 1000 km. The bad cases typically occur when the target is far from any landmarks. Figure 5(a) shows the relationship for the University dataset using SOI (the results are very similar for CBG); the graph shows a definite trend toward error increasing almost exactly with the distance to the nearest landmark. Geolocation rarely works much better than this distance; only twice does the estimation error beat the distance to the nearest landmark by more than 100 km. Figure 5(b) shows the correspondence between the smallest latency measured to a target and the estimation error for that target. When the latency is small, a landmark is nearby and the feasible region for the target is small, so any estimate will be good and any technique should work. The median error for the 53 targets with a latency less than 4 ms is 15 km; for the 75 with no latencies less than 4 ms, the median is 266 km. This observation also suggests why Shortest Ping is competitive with more complicated techniques.
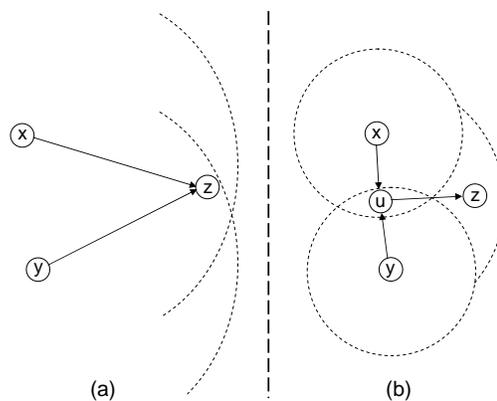
Figure 6: Shared routers indicate indirect paths.

3.6 Conclusions

We give three high-level takeaway points about delay-based geolocation:

- At least with a homogeneous set of landmarks (and most regional deployments within a single administrative domain are likely to be homogeneous), inter-landmark measurements and per-landmark bestlines may not gain CBG anything versus much simpler techniques.

- The distance from a target to the nearest landmark strongly predicts the estimation error. Therefore, delay-based techniques only provide consistent quality if landmarks are ubiquitous.

- The shortest round-trip time to a target is a good empirical predictor for the error in delay-based techniques. Delay-based techniques work well when the shortest RTT is small, and they often work poorly when it is not small.

Therefore, we seek techniques that can perform well even when no landmark is within a small delay of the target.

## 4. TOPOLOGY BASED GEOLOCATION

We begin this section by motivating the need for taking network topology into account during geolocation. We make a series of observations based on specific examples and constructed topologies, with each observation followed by the corresponding design implication that a geolocation technique needs to address.

Challenge: CBG bestline constraints attempt to compensate for the fact that Internet paths are sometimes geographically circuitous and sometimes inflated. Unfortunately, the directness of a network path from a landmark to a particular target cannot be predicted a priori; a single conversion factor for the entire network or even a per-landmark conversion factor is not sufficient to capture the intricate details of the network topology and routing policy.

Consider for instance the two cases illustrated in Figure 6. In the topology on the left, landmarks x and y attempt to geolocate target z to which they have direct one-hop network measurements. The position of the target is then estimated to be at the intersection region of two circles centered around the landmarks. The topology on the right has the landmarks encountering a common router u on paths to the target, making the paths less direct. In this case, the geolocation technique should take into account that the shared router indicates less direct end-to-end paths and the resulting position estimate for z should be closer to x and y than what the end-to-end measurements would indicate.
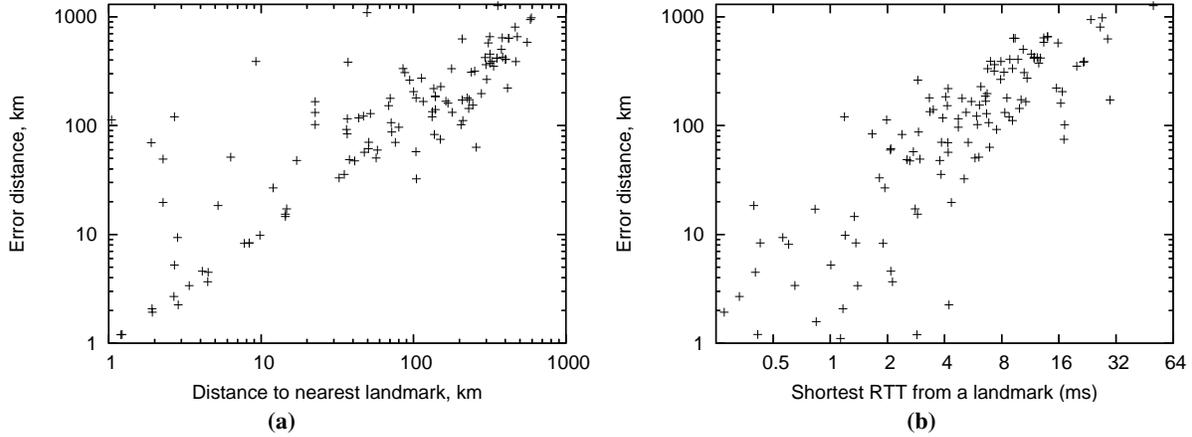
Figure 5: (a) SOI location error as a function of the distance to the nearest landmark. (b) SOI location error as a function of the shortest ping latency measured from a landmark.
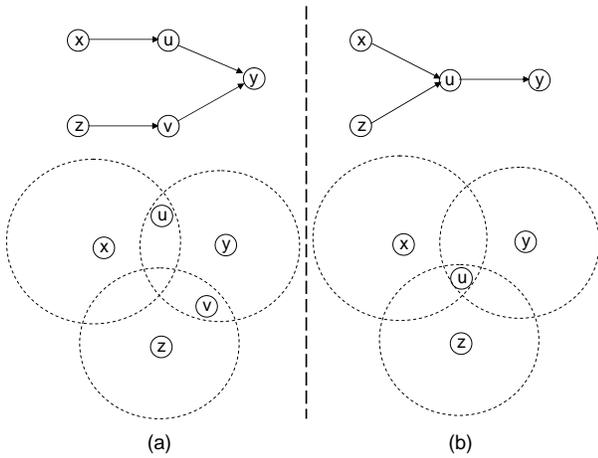
Figure 7: Router aliases: Accuracy improves from (a) to (b) as v is identified as an alias for u.

Design Implication: A geolocation technique has to therefore take network topology and routes into account in order to capture path-specific latency inflations.

Challenge: The constrained multilateration techniques work well when the target is close to one of the landmarks. This observation extends trivially to routers near landmarks. For instance, in the topology under consideration in Figure 6, one might be able to accurately geolocate the router u using direct one-hop measurements from x and y. It is also possible that once u is geolocated accurately, it can then serve as a landmark for geolocating other network entities. Imagine however that the router is one hop away from x but multiple hops away from y. Then the process of geolocating u might require the position estimates of other routers, whose position estimates are in turn affected by u's position.

Design Implication: In order to achieve a consistent and more accurate solution, a geolocation technique has to simultaneously geolocate the targets as well as routers encountered on paths from landmarks to other landmarks or targets.

Challenge: One can geolocate intermediate routers using hop-to-

hop measurements only if there are a sufficient number of constraints on the router. A naive attempt might miss crucial structural constraints, leaving the system under-constrained. For instance, the left hand side of Figure 7 depicts the process of geolocating intermediate IP addresses u and v that are thought to be separate. If we identify that u and v are two different network interfaces on the same router, the feasible region in which the corresponding router could be located becomes smaller, shown on the right.

Design Implication: To tightly fix the feasible locations of routers, a geolocation technique must use measurements to extract existing structural constraints, including by identifying collocated interfaces.

Challenge: There are inherent limitations to purely measurement-based techniques even if network topology is taken into account. Consider for instance academic institutions in Alabama serviced by the Gulf Central GigaPoP (depicted in Figure 8). When the landmarks are located outside this network, all routes to targets in this network pass through "Southern Crossroads" (SOX), a connection point in Atlanta that links to the Abilene national network. The network structure could be used to geolocate the SOX routers, but any attempt to geolocate the Gulf Central GigaPoP nodes will be foiled by the lack of structural constraints in this stub network with a tendril-like structure.[3]

The only avenue to improving accuracy in such settings is to incorporate passive landmarks and also network elements whose probable locations could be obtained from GPS, DNS rules or other databases. The active landmarks can make network measurements to each other and to the passive landmarks and targets; the intersections of routes to fixed locations with routes to the target help constrain the system. However, inaccurate location hints could throw off location estimates.

Design Implication: A geolocation technique must use delay and topology measurements to validate the locations of passive landmarks and external location hints and then incorporate them to overcome insufficient structural constraints.

Based on the above set of observations, we now present an approach that takes network topology into account while geolocat-

---

[3]Heuristics such as using population densities could also result in substantial errors by mapping all Alabama targets to the most populous city in Alabama.

Figure 8: Gulf Central GigaPoP. The tendril-like Alabama network lacks sufficient structural constraints.

ing end-hosts. We call our approach Topology Based Geolocation (TBG). TBG is an extension of constrained multilateration. It augments delay measurements to end-hosts with information about topology and routing and performs a form of constrained multilateration on all of the intermediate routers and the targets simultaneously. We begin by outlining the basic approach, and then address issues related to clustering network elements and validating other sources of information, before presenting details on the constrained optimization. A summary of techniques used and a roadmap for the approach is provided in Table 1.

4.1 Overview of Proposed Technique

Our primary tool in obtaining network topology information is the `traceroute` tool. The landmarks issue `traceroute` probes to each other and the target. This provides us with round-trip measurements to the target and the identity of the intermediate network interfaces. The differences between the RTTs to adjacent network interfaces provides estimates for link latencies. We also identify network interfaces that are collocated. The entire set of traceroutes from landmarks to other landmarks and from landmarks to the target, when combined with structural observations about collocated interfaces, provides us with the network topology.

Using the end-to-end delays, inferred per-hop latencies, and the topology, we attempt to geolocate the target along with all of the intermediate routers using a constraint-based optimization technique. The goal is simply stated. We want to find positions for the target and the routers such that the distance between the positions of every pair of adjacent network elements is proportional to the corresponding link latency measurements. When the topology is sufficiently well-connected, there will be multiple constraints on the positions of intermediate routers, forcing them to be placed close to their actual positions. The resulting placements then model the geographical detours taken by end-to-end paths, thereby providing a quality estimate for the target's position.

4.2 Network Topology Generation

We now address various issues related to data gathering, the accuracy of network measurements, and the ability to generate a topology that can be used as an input by our topology-based techniques.

4.2.1 Hop Latency Estimates

We use `traceroute` measurements to infer link latency. Hop latency estimates could be obtained by using the difference in round-trip times to adjacent routers. Unfortunately, the hop latency estimates are accurate only if the link is traversed both in the forward

and reverse directions, a property that might be violated if routing is asymmetric. We use three different techniques to address this issue.

First, we can discard some egregiously inflated measurements by taking into account the reverse TTL values observed on probe replies from intermediate routers. We can often determine the reverse path hop-length from an intermediate router, as most routers initialize the time-to-live values for their packets from a small number of possibilities. (There are six common initial TTL values: 30, 32, 64, 128, 150, and 255.) If the estimated hop count of the reverse path changes significantly from one forward node to the next, we discard the link estimate between the nodes as likely to be flawed. Similarly, as measurements of MPLS segments could be subject to inflation, we use measurements to simply upper-bound the geographical distance between routers using MPLS.[4]

Second, we measure paths in both directions between pairs of landmarks and if both the forward and reverse paths traverse a particular link, then we are guaranteed that the hop latency estimate obtained by taking the differences of measurements to the two endpoints is indeed accurate. We can attach a high confidence factor to those measurements, and the link estimates can be favored over others during geolocation.

Third, we can increase the number of vantage points from which we probe a certain link in order to obtain more samples for estimating the link latency. For every link observed on a path from a landmark, we issue probes to both its endpoints from all of the other landmarks. If these probes pass over the link, then they can provide an independent estimate for the link. We then generate a confidence factor associated with this link delay using the variance of the measurements and the number of independent vantage points that were able to probe the link.

4.2.2 Clustering

As mentioned earlier, clustering interfaces that belong to the same router (IP aliases) leads to better constrained router locations. In addition, we can also obtain better estimates for the link latency after clustering as there are likely to be more observations of paths traversing each induced link.

We identify IP aliases using two existing techniques, Mercator [9] and Ally [24]. The Mercator technique works as follows. UDP probes are sent to high-numbered ports on a set of network interfaces. Routers typically send back a port-unreachable ICMP message with the source address of the output interface on the router. If two probes to two different interfaces elicit replies with the same source address, then they are clearly aliases.

Ally, on the other hand, targets routers that do not have the above feature. The Ally technique has to be used on pairs of interfaces. Since it would be expensive to run the Ally technique on all pairs of interfaces, we first generate candidate pairs that are likely to be aliases using the following two methods. First, we probe all of the interfaces from a vantage point and generate candidate pairs that have similar reverse path lengths for their probe replies. Second, we build a traceroute graph of all paths from all of the landmarks and select pairs of interfaces if they share a common interface as the next hop. As with the reverse path length heuristic, this heuristic is also designed to identify interface pairs that are nearby in the Internet topology.

Given a candidate alias pair, Ally sends probes to the two interfaces and examines the IP identifier (IP-ID) field of the responses. Most routers generate the IP-ID values using a single counter that

---

[4]In future work, we believe that we can further constrain MPLS routers using IP route recording [22] and the transition links between MPLS and non-MPLS networks, likely to be short hops.

Technique	Description	Goal	Section
traceroutes from landmarks	Landmarks issue ICMP-based traceroute probes to targets, identify intermediate interfaces, and per-hop latencies.	map topology	Section 4.1
estimate hop latency	Generate confidence factors for hop measurements by checking for symmetric paths, examining TTL drops, and computing variance.	improve measurement accuracy	Section 4.2.1
cluster network interfaces	Identify network interfaces that are geographically collocated by checking for network aliases.	increase structuring	Section 4.2.2
validate location hints	Validated and incorporate information about passive landmarks and DNS location hints.	incorporate location hints	Section 4.2.3
constrained optimization	Solve for locations of targets and intermediate routers using all of the above topology information.	geolocate targets	Section 4.3

Table 1: A summary of techniques used in Topology-Based Geolocation.

is incremented after each packet has been generated. By checking for the nearness of the IP-ID values and by performing repeated checks at different points in time, one can ascertain whether two interfaces are aliases with a high degree of certainty.

4.2.3 Validating Location Hints

Certain routers and end-hosts have DNS names that could be parsed to provide hints regarding where they are located, typically airport codes or abbreviated city names as part of the name [24, 28]. For example, *bos-edge-03.inet.qwest.net* is likely in Boston. The issue with using router DNS name information is that they could be misnamed, with the naming errors introduced by router reconfiguration, router repairs/upgrades, and reassignment of IP addresses [29]. Some names may be correct but misleading or ambiguous: does *chrl* indicate Charlotte, NC, or Charleston, SC? Any geolocation technique that uses location hints must validate the location derived from DNS parsing rules before using it to geolocate the target. Fortunately, TBG is well-placed to perform this validation as there are a rich set of topology constraints that can be used to verify location hints.

First, we can use RTT measurements from landmarks to intermediate routers to rule out some locations; if the RTT along a path to a router with a predicted location violates the speed of light, then we invalidate the location hint. Second, clustering enables us to identify inconsistencies and potentially correct misnamings. When the location hints for a majority of interfaces in a cluster point to a certain geographic location, we could attribute this majority value to the entire cluster. Third, for those location hints that survive the previous two checks, we use the observed network topology and measured hop latencies to verify them. We use the following simple technique. Each landmark and each router interface, if it has an associated location hint, votes on the location hints for adjacent network interfaces to which it is connected. It casts a positive vote for those interfaces whose location estimates are consistent with the hop-by-hop measurements and a negative vote otherwise. Interfaces that accumulate sufficient positive votes are deemed to be properly named and their location hints are validated. If the location hint is validated, then we pin down the router's location, meaning we consider the router as a passive landmark in our network topology.

As part of a feasibility study, we generated ISP-level maps for the five largest ISPs in the US: Level3, Uunet, Sprint, AT&T, and Qwest. We observed a total of 23720 IPs in these networks, of which 8229 resolved to a location using undns rules. 6493 IPs at 208 distinct locations passed the speed of light test, and 3561 at 131 distinct locations also passed the vote test, which was performed using stringent thresholds. This represents a substantial increase in the number of network elements that could be used as landmarks if they are observed in routes to targets.

4.3 Constrained Optimization

We model the network as a graph with n landmarks with known positions and m targets with unknown positions. Any intermediate routers with unknown positions are also considered as targets in this formulation. The goal is to compute the positions of the targets $X = [x_1, x_2, \ldots, x_m]$, given the positions of the landmarks $L = [l_1, l_2, \ldots, l_n]$. One or more of these targets could be end-hosts that need to be geolocated, with the remaining targets being routers encountered on paths. We denote the distance between i and j by $d(i, j)$.

We now discuss the types of constraints measurements place on the distances and locations.

Hard Delay Constraints: If we have a round-trip delay between landmark l_i and target x_j, the distance between the two nodes is bounded by the distance c_{ij} that light can travel through fiber in that period of time. Therefore, we can impose a hard constraint, one that cannot be violated, and represent it as:

$$d(l_i, x_j) \leq c_{ij}.$$

We will refer to the set of hard delay constraints as C_d. By including these constraints, every TBG estimate is within the feasible region defined by the speed of light. So, we can think of TBG as using strictly more information than techniques such as CBG and SOI, which only define a region.

Soft Link Latency Constraints: In Section 4.2.1, we describe how we estimate the latency of a link. A hop latency measurement between two adjacent nodes could be used to provide a distance estimate h_{ij}. Since these measurements are typically noisy and since the distance estimate does not account for issues such as the fact that fiber is not laid as the crow flies, we represent these as equality constraints with an error term that has to be minimized during geolocation: the node placement should respect the predicted distance as closely as possible. We can represent one of these constraints as:

$$d(x_i, x_j) = h_{ij} + e_{ij}.$$

e_{ij} will appear in the minimizing objective function, so will be driven down to zero if the distance between x_i and x_j matches h_{ij}.

We will refer to this set of link constraints as C_l.

Optimization Problem: We assume that the geographical coordinates composed of the latitude and longitude values are projected into the plane $\mathbf{R}^2$, as with a map. We can solve for the positions of targets, X, while minimizing the total amount by which the hop distance estimates are violated. Minimizing this total places nodes such that the realized inter-node distances deviate as little as possible from the actual measurements.

$$\text{minimize}: \quad \sum_{i,j \in C_l} |e_{ij}|$$
$$\text{subject to}: \quad C_d, C_l.$$

The above problem is not a convex optimization problem, but we follow a formulation from the sensor network community to recast the problem as a semidefinite program [25], thereby allowing us to use to fast solvers (such as SeDuMi [21]) to perform the optimization. One could also use a modified form of Vivaldi [6] to solve for the coordinates of the routers and the end-hosts given the topology and delay constraints. We chose not to do so because our experiments with a straightforward implementation of Vivaldi sometimes failed to converge to a global optimum for large, complex Internet maps.

In practice, we modify the optimization problem described above in order to address the following issues. First, as with most constraint-based optimization problems, we would like to minimize both the absolute and relative deviations between actual measurements and the program solution's realizations. For instance, a 1 km error for a 10 km estimated link distance is significantly more troublesome than a 1 km error for a link estimated to be 100 km. Second, some of the measurements are less reliable than others, an issue that we discussed in greater detail in Section 4.2.1; it would be meaningful to associate a linear confidence factor with each measurement and use this factor to scale the appropriate error variables in the optimization function. Third, we use a stricter form of hard end-to-end constraints than described above. The hard delay constraints mentioned above do not take into account the geographic circuitousness of paths. Instead of just requiring the end hosts to be no further away than a distance determined by the measured delay, we require that the length of the path through the placed routers along the route be no more than this distance. Note that the end-to-end constraints would drive the sum of the small deviations for individual link constraints down to zero. All of these issues can be easily captured in our formulation.

4.4 Mapping the Target to the Last Constrained Router

Most end-hosts are not tightly constrained, with all routes to a given host sharing the same last few hops. TBG is left with many possible positions that could explain the measurements past the last constrained router. This limitation is fundamental to any technique that seeks to perform automatic IP geolocation. We address the limitation by using the location of the last constrained router as our location estimate. This approach is analogous to CBG's use of the centroid of the intersection region, but on a smaller scale. Just as the size of the intersection region gives an idea of CBG's confidence in an estimate, the hops from the last constrained router gives an idea of TBG's. Fortunately, the final hops are usually short and the estimation error is consequently minimal. This approach also allows TBG to provide quality estimates for hosts behind edge-firewalls/NATs that filter out ICMP requests, as geographically-close routers along paths to the targets still reply. Further study is necessary, but, upon informal investigation of more than 100 targets that did not reply, TBG seems to provide similar quality results regardless.

5. EVALUATION

In this section, we demonstrate that TBG can be used to improve estimates versus delay-based techniques. We first give three variants on TBG. Then in Sections 5.2 and 5.3, we evaluate TBG in settings where the network topology is sufficiently rich to enable geolocation without the need for passive landmarks. We consider targets in both academic and non-academic networks for these experiments. In Section 5.4, we evaluate TBG on our University dataset, which contains many targets in stub networks, and study its accuracy with and without passive targets. Lastly in Section 5.5, we evaluate the effectiveness of TBG using only a single landmark to probe the targets, with the rest being passive. In an actual deployment, we envision a full set of landmarks tasked with mapping out the Internet, with a subset probing on-demand to geolocate a new IP address; such a tiered deployment reduces the probing overhead. This last experiment, with only a single probing landmark and the rest simply checking for aliases and validating hints, represents the limit of such an approach and still provides results competitive with earlier techniques.

5.1 TBG Variants

In this section, we present three settings for applying our constrained optimization approach to the topology-based geolocation problem. The three settings use the same optimization framework and only vary by which nodes are part of the landmark set L.

Active Landmarks only: In this setting, the only locations assumed to be known are those of the probing landmarks. We refer to this variant below as *TBG-pure*, because it assumes no other information than the locations of the probing hosts.

Active and Passive Landmarks: In this setting, in addition to the locations of probing landmarks, there is a set of passive landmarks with known locations, to which measurements can be made but which cannot originate any measurements of their own. We refer to this variant as *TBG-passive*.

Active and Passive Landmarks, plus Validated Hints: In this setting, in addition to the active and passive landmarks, the program is given the hinted locations of a set of routers, validated according to the techniques described in Section 4.2.3. We refer to this variant as *TBG-undns*.

5.2 Abilene Dataset

In this section, we examine the performance of TBG-pure in geolocating the PlanetLab hosts collocated with Abilene's 11 PoPs (see Figure 9). For geolocating each target, we use the remaining 10 PlanetLab nodes as the landmarks. In this evaluation, we do not use passive landmarks nor do we rely on DNS naming conventions. This dataset is simple enough for us to gain insights into the strengths and weaknesses of delay and topology-based techniques.

Figure 10 shows the error in estimation for TBG-pure, SOI, and CBG on these targets. TBG-pure has a mean error of 209 km while SOI and CBG have mean errors of 334 km and 361 km respectively. Furthermore, TBG-pure's maximum error is only 325 km while SOI and CBG have maximum errors in excess of 1100 km. More detailed analysis of the results illustrates two important points.

First, the centroid of the intersection region from a set of circles cannot lie outside the convex hull of the centers of the circles. So CBG always estimates targets to lie within the convex hull of its landmarks, limiting its ability to geolocate targets outside that hull. TBG can geolocate targets outside the hull, given enough constraints. For instance, CBG gives an error of about 400 km for the PlanetLab target at the Atlanta PoP, as it draws the target into the convex hull of the landmarks, placing it roughly halfway between Atlanta and Kansas City. TBG-pure has a much smaller error of about 100 km.

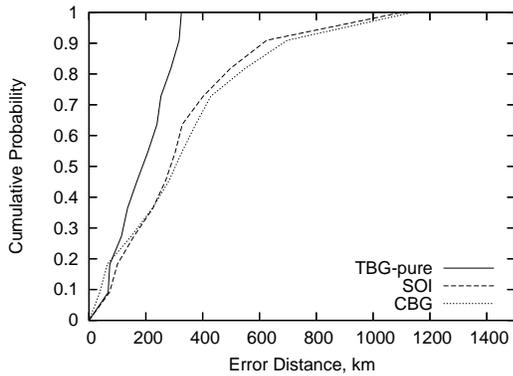**Figure 9: PlanetLab nodes collocated with Abilene PoPs.**

Figure 11: Estimation errors for Sprint dataset.

Figure 10: Estimation errors for the 11 nodes in the Abilene dataset.

Second, TBG-pure's errors are not due to imperfect measurements,[5] but are due to underconstrained targets (e.g. New York in Figure 9) and the discrepancies between fiber lengths and geodesic distances. Delays represent the length of fiber laid and not the "as the crow flies" distance, which means that the observed end-to-end speed varies enough to distort multilateration. We observe that fiber often follows transportation corridors and therefore driving distances often correlate with hop latencies. As an example, the Denver Abilene PoP observes end-to-end speeds of 60 km/ms, 63 km/ms, and 82 km/ms to Sunnyvale, Seattle, and Kansas City, respectively, making accurate triangulation impossible. However, the delays would correspond to the more consistent speeds of 84 km/ms, 83 km/ms, and 89 km/ms over the driving routes between the cities [14]. But, as the end-points of a link are not known apriori, a measurement technique cannot incorporate per-link conversion factors. We also note that these errors could be minimized if there are more constraints on each router.

5.3 Sprint Dataset

We next evaluate the techniques for targets in non-academic networks, but still in a setting that provides sufficient structural constraints. We attempt to geolocate routers in the Sprint network using our 68 PlanetLab landmarks. We consider 22 targets collocated with Sprint PoPs in the US, including representative IPs from every Sprint PoP listed in the Sprint Looking Glass server [16]. As with the Abilene experiment, we do not make use of passive landmarks

---

[5]Our delay measurements of backbone links in the I2 setting (as well as over other ISPs) had remarkably low variance.
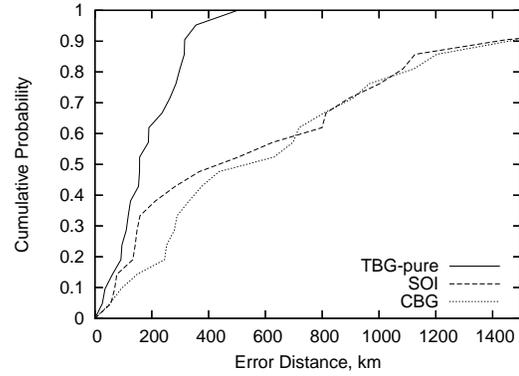
or other hints regarding location information.

We make two observations regarding this experiment. The Sprint backbone has a richer network structure than the Abilene network, thereby providing more information to topology-based multilateration techniques. On the other hand, probes to targets in the Sprint network could be subject to path inflation and other complications arising out of more complex inter-domain routing policies.

The results of the experiment are depicted in Figure 11. TBG-pure has a mean estimation error of 194 km, while SOI and CBG have mean errors of 689 km and 749 km respectively. The high estimation errors of the CBG variants are due to increased levels of path inflation, which in turn reduces the correlation between end-to-end distances and end-to-end RTTs. For instance, even though we had a landmark in New York City, its probe to the Sprint target in New York City was routed through routers in Washington, DC and Ashburn, VA before reaching the target, resulting in a RTT of 15 ms. The CBG-based techniques incur errors of 246 km and 288 km for this target. TBG-pure is able to locate the target with an error of only 61 km, as it is able to take into account the circuitousness of paths to these non-academic targets while estimating their locations. As many real-world targets are likely to have at least partially inflated paths, this experiment illustrates how TBG is able to overcome the inherent limitations of pure delay-based techniques.

In the Abilene and Sprint datasets, even without location hints or passive landmarks, TBG harnesses the structured topologies to reduce the average error by 40% and 70%, respectively, and the 90th percentile of error by a factor of 4.

5.4 University Dataset

We then evaluate topology-based geolocation techniques on our University dataset. In Section 3, we gave results that none of the delay-based techniques provide consistently good estimates when the shortest round-trip time measured from a landmark to the target exceeds 4 ms. In this section, we evaluate whether TBG techniques can improve the location accuracy for these difficult targets.

As mentioned earlier, many of the education institutions reside in tendril-like stub networks that lack sufficient structural constraints. Consequently, a pure topology-based scheme is unlikely to provide good estimates; it can only accurately geolocate the last constrained router on the paths to the target. So in addition to evaluating TBG on this dataset, we also evaluate TBG with passive landmarks (TBG-Passive), where we geolocate each target assuming that the rest of the target set are passive landmarks.

We also evaluate the accuracy of TBG after incorporating location hints that have been validated. We refer to this variant as
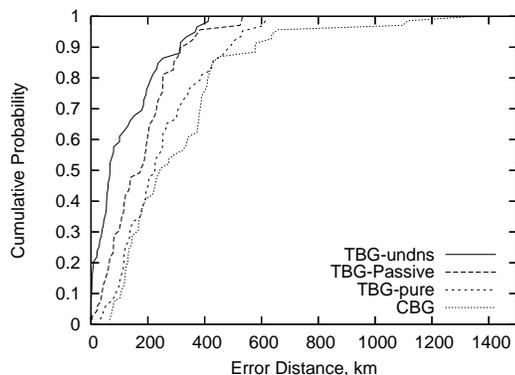
Figure 12: Estimation error for University targets.

Figure 13: Estimation error, CBG probing from all landmarks, TBG probing from one using validated location hints.

TBG-undns. We have 8,321 interfaces in our University dataset, of which 7,164 have DNS names. We did not use location hints for any *.edu*'s. Excluding *.edu*'s, parsing rules gave location hints for 5,509 interfaces, representing about 200 cities in 42 states plus Washington, DC. 7 of these interface locations were discarded for violating the speed-of-light by measurements from landmarks, 17 locations were invalidated by IP alias information.

Before we present the evaluation results, we provide statistics on the topology generation process. We were able to identify 1,672 alias pairs using the Mercator technique and an additional 621 alias pairs using the IP-ID based scheme. Furthermore, our probes traversed a total of 135,489 network hops, of which only 1,392 were not responsive. We note that we used ICMP probes that most routers in the Internet backbone respond to. We had observed that about a quarter of the backbone routers do not respond to UDP probes, and hence used ICMP probes.

Figure 12 compares the estimation errors of TBG, with various added information, with that of CBG. We do not include GeoPing because, as shown in Section 3.5, even with passive landmarks it does not perform as well as CBG. We make the following observations. As expected, pure TBG does not greatly outperform CBG on this dataset, but the use of passive landmarks and validated location hints improve the accuracy. CBG has a much longer tail than the TBG variants, just as with the results in the Abilene and the Sprint datasets. The lack of topology information means that CBG is not able to take into account the circuitousness of paths. It simply ends up incurring large intersection regions for targets with moderate to high latencies from the set of landmarks. TBG, though not perfect in this less structured setting, is at least capable of constraining the last backbone router en route to the target. TBG-Passive and TBG-undns are able to improve the estimation errors by constraining routers that are closer to the target by using passive targets and location hints. Overall, the mean estimation errors for TBG-undns, TBG-Passive, TBG, and CBG are 138 km, 178 km, 253 km, and 296 km, respectively. The median estimation errors for the same list of techniques are 67 km, 176 km, 225 km, and 228 km. Using verified location hints to enhance topology information in stub networks, TBG-undns improves the average error by 53%, the 90th percentile of error by 35%, and the worst-case error by a factor of 3, as compared to CBG.

5.5 Single Probing Landmark

We evaluate TBG-undns using only a single active landmark, with the rest serving as passive landmarks, and we compare to CBG (which still uses all active landmarks). We want to discern how ef-
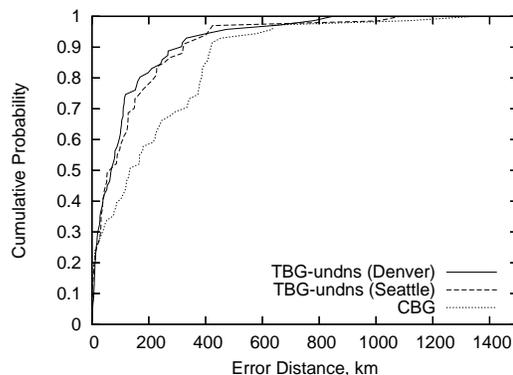
fectively TBG can leverage passive location information with only a single probe to the target. We use the University dataset from Section 5.4, with two slight modifications. First, we note that the single landmark is unlikely to be near many targets, so we no longer exclude targets with measured delays less than 4 ms. Therefore, we can evaluate how effectively TBG can estimate locations when it does not have a nearby probing site, but CBG does. Second, for our evaluation, we use as our single landmark a PlanetLab host collocated with an Abilene router. The Abilene network does not carry traffic to certain prefixes, so we are only able to make measurements to 67 targets and evaluate both CBG and the single-probe TBG-undns only on them.

In this setting, TBG-undns is given the locations of all landmarks and verified router locations, then the active landmark makes a `traceroute` measurement to each target, as well as to each of the passive landmarks and location-verified routers. Figure 13 shows the results when the single landmark is the Seattle I2 PlanetLab host and when it is the Denver I2 PlanetLab host, as compared to CBG. From Seattle, TBG-undns has a median error of 68 km and a mean of 136 km, and Denver yields a median estimation error of 68 km and a mean of 130 km. CBG has a median of 154 km and a mean of 235 km. Even probing just from a single vantage, we observe that TBG makes good use of the topology information and location hints, and is able to provide better estimates than CBG for most targets.

6. DISCUSSION

By performing constrained multilateration at the router level and by leveraging passive landmarks, TBG can be viewed as a technique that combines the best attributes of the various delay-based techniques and employs them at a finer granularity (router level as opposed to end-host level) than the delay-based techniques, allowing it to overcome the weaknesses of delay-based techniques.

CBG and its variants perform well when the target is close to one of the landmarks. However, the delay-based techniques do not perform well where the targets are not close to the landmarks and where the geographical circuitousness of paths to the target could not be modeled. These two factors imply that CBG rarely performs well without a short delay from some landmark to the target. Further, these techniques require landmarks that completely encircle the target, as all of their predicted positions lie within the convex hull induced by the landmarks.

TBG addresses these issues by taking network topology into consideration. TBG is a form of constrained multilateration, but

it performs the constraint-based location on all of the routers and the target simultaneously, with the eventual solution providing a placement for all of the network elements in a manner that is most consistent with the network measurements. Consequently, as illustrated by the results on the Abilene and Sprint datasets, TBG can work well even where the landmarks are far from the target or where there is substantial path inflation, whereas purely delay-based techniques incur significantly higher estimation errors in such settings. TBG can also geolocate targets outside the convex hull formed by the landmarks as long as there are sufficient constraints on their positions, as the constraints are best satisfied by pushing the target away from the central region.

However, TBG has an algorithmic limitation. TBG produces quality geolocation only if there are sufficient structural constraints on the target. As discussed, this limitation can be overcome by using of passive landmarks and validated location hints. The idea of passive landmarks comes from GeoPing. Passive landmarks and location hints aim to induce constraints on routers close to the target, allowing TBG to use location information and constrain stub networks without an active probe inside them.

Our technique, however, incurs a higher measurement cost than simple delay-based techniques such as CBG and GeoPing. Our proposed solution requires the use of `traceroute` probes to discover the network structure. In addition, we use targeted probes on pairs of network interfaces to determine aliases.

Fortunately, most of these measurements could be performed ahead of time before we geolocate any particular target, and therefore the cost could be amortized over many target geolocations. The probes between landmarks, the task of identifying aliases encountered on these inter-landmark paths, and the geolocation of the corresponding routers can all be performed once and reused for different targets. The incremental measurement cost of geolocating a new target is in issuing one `traceroute` probe from each of the landmarks to the target and then possibly generating and checking a few new alias candidates, an optional task that is required only if there are not enough constraints to place the routers. With the relative effectiveness of TBG probing from a single vantage point as compared to CBG, a user could conceivably obtain our dataset of topology information, make additional probes from only a single machine to new targets, and hope to get results at least comparable to CBG, which requires additional probes from many vantages. We note that a geolocation service must periodically refresh information regarding the network topology and perform on-line checks for staleness of topology information, all of which are engineering tasks that would have to be addressed by such a service.

7. CONCLUSION

In this paper, we studied automatic techniques for IP Geolocation. We began by studying the performance of existing techniques that rely solely on delay measurements. We proposed simpler variants that performed as well as the existing techniques, but all of the delay-based techniques had fundamental limits on accuracy while geolocating targets that are distant from the landmarks. We therefore proposed a new technique, Topology Based Geolocation, which extends previous constrained multilateration techniques, by using topology information, generating a richer set of constraints, and applying a more sophisticated optimization technique to derive the placements. Unlike the previous state-of-the-art, TBG incorporates and validates network router location hints and passive landmarks to achieve accurate estimates even for underconstrained, distant targets. We show that, using this extra information, TBG outperforms existing techniques on a real world dataset, especially improving estimates for the difficult cases. On our university-

based dataset, TBG generates estimates that are twice as good on average and more than three times better for the hardest targets. By highlighting the limitations of existing techniques and the challenges of measurement-based geolocation, as well as giving a new technique to deal with these limitations, we have laid a groundwork upon which further experiments and topology-based techniques can build.

8. REFERENCES

[1] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, Tel-Aviv,Israel, 2000.

[2] S. Banerjee, T. Griffin, and M. Pias. The interdomain connectivity of PlanetLab nodes. In *Proceedings of Passive & Active Measurement*, 2004.

[3] E. G. S. Bernard Wong, Aleksandrs Slivkins. Meridian: A Lightweight Network Location Service without Virtual. In *ACM SIGCOMM*, 2005.

[4] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of Information Processing in Sensor Networks*, 2004.

[5] D. Clark, C. Partridge, R. Braden, B. Davie, S. Floyd, V. Jacobson, K. Kitabi, G. Minshall, K. Ramakrishnan, T. Roscoe, I. Stoica, J. Wroclawski, and L. Zhang. Making the World (of Communication) a Different Place. *ACM SIGCOMM Computer Communication Review*, 35(3), 2005.

[6] F. Dabek, R.Cox, F.Kaashoek, and R.Morris. Vivaldi: A decentralized network coordinate system. In *Proc. of the ACM SIGCOMM*, Portland, OR, USA, 2004.

[7] L. Doherty, K. S. J. Pister, and L. E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of Infocom*, pages 1655–1633, 2001.

[8] P. Enge and P. Misra. Special issue on global positioning system. In *Proceedings of the IEEE*, volume 87, pages 3–15, jan 1999.

[9] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *Proc IEEE Infocom*, 2000.

[10] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of Internet hosts. *To appear in ACM Transactions on Networking*.

[11] http://geonames.usgs.gov. Geographic Names Information System.

[12] http://maps.google.com/. Google Maps.

[13] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device positioning using radio beacons in the wild. In *Proc. of Pervasive*, Munich,Germany, 2005.

[14] maps.yahoo.com/dd. Yahoo! Driving Directions.

[15] D. Moore, R. Periakaruppan, J. Donohoe, and K. Claffy. Where in the world is netgeo.caida.org? In *Proc. of the INET*, Yokohama,Japan, 2000.

[16] oxide.sprintlink.net. Sprint Looking Glass Server.

[17] V. N. Padmanabban and L. Subramanian. Determining the geographic location of Internet hosts. In *SIGMETRICS/Performance*, pages 324–325, 2001.

[18] V. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *Proc. of the ACM SIGCOMM*, San Diego,CA,USA, 2001.

[19] R. Percacci and A. Vespignani. Scale-free behavior of the Internet global performance. *The European Physical Journal*

B - Condensed Matter, 32(4):3–15, apr 2003.

[20] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In *Proceedings of MOBICOM*, Boston,MA,USA, 2000.

[21] sedumi.mcmaster.ca. Sedumi.

[22] R. Sherwood and N. Spring. Touring the Internet in a TCP sidecar. In *ACM SIGCOMM*, 2006.

[23] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, Aug. 2003.

[24] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, 2002.

[25] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1), 1996.

[26] ws.cdyne.com/ip2geo/ip2geo.asmx. IP2GEO Website.

[27] www.netgeo.com. Netgeo Website.

[28] www.sarangworld.com/TRACEROUTE/. Sarangworld Project.

[29] M. Zhang, Y. Ruan, V. Pai, and J. Rexford. How DNS misnaming distorts Internet topology mapping. In *Proc. of USENIX Annual Technical Conference*, 2006.