Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts

Bernard Wong, Ivan Stoyanov, Emin Gün Sirer
Dept. of Computer Science, Cornell University, Ithaca, NY 14853

Abstract

This paper outlines a novel, comprehensive framework for geolocalization, that is, determining the physical location of Internet hosts based on network measurements. The core insight behind this framework is to pose the geolocalization problem formally as one of error-minimizing constraint satisfaction, to create a system of constraints by deriving them aggressively from network measurements, and to solve the system using cheap and accurate geometric methods. The framework is general and accommodates both positive and negative constraints, that is, constraints on where the node can or cannot be, respectively. It can reason in the presence of uncertainty, enabling it to gracefully cope with aggressively derived constraints that may contain errors. Since the solution space is represented geometrically as a region bounded by Bezier curves, the framework yields an accurate set of all points where the target may be located. Preliminary results on PlanetLab show promise; the framework can localize the median node to within 22 mi., a factor of three better than previous approaches, with little error.

1 INTRODUCTION

Determining the physical location of Internet hosts, known as *geolocalization*, is a building block and critical enabler for a wide range of services that depend on knowledge of a computer's physical location. Accurately determining the position of a node in the real world based solely on network measurements, however, poses many challenges. The key obstacles to accurate and precise geolocalization comprise how to represent network locations for nodes, how to extract constraints on node location from noisy Internet measurements, and how to combine these constraints to yield good estimates of node position.

In this paper, we present a novel and comprehensive framework called Octant for geolocalizing hosts on the Internet. Octant provides a general framework which represents node positions precisely using Bezier-bounded regions, expresses constraints succinctly as areas, and computes positions accurately by solving a system of geometric constraints. The constraint system is anchored to the physical globe using a small number of landmarks whose positions are approximately known. The Octant approach is comprehensive and general; it enables almost all past work on geolocalization to be expressed within the framework, as a (limited) subset of the techniques described in this paper.

Octant represents the potential area where a node can be located as a surface bounded by Bezier curves. The Bezier curve representation is general; the enclosed area may be non-convex and even consist of disconnected regions. The areas are expressed in a compact manner, and boolean operations on areas such as union, intersection, and subtraction are computed efficiently. These properties enable Octant to admit and cohesively use *positive information*, that is information on where the node may be located, as well as *negative information*, information on where the node does *not* reside. The use of both positive and negative information contrasts with past approaches that rely solely on positive information, and accounts for the increased generality and accuracy of the Octant framework.

Octant uses various principled methods to extract precise constraints from noisy Internet measurements. It compensates for dilation stemming from queuing delays by computing an extra "height" dimension that captures the queuing effects. It minimizes the impact of indirect routes through piecewise localization of routers on the network path, where it localizes ordinary routers on the path and uses their approximate location to further refine the position estimate of the target node. It can integrate additional data from the WHOIS database, the DNS names of routers, and the known locations of uninhabited regions to refine the solution. Finally, Octant uses a weighted solution technique where weights correspond to confidence in a derived constraint to enable the use of aggressive constraints in addition to conservative ones without creating a non-solvable constraint system.

We have implemented and deployed a preliminary version of our system, using some PlanetLab [3] nodes as landmarks. Measurements show that Octant achieves a median error of 22 miles for its position estimates, compared to 70 miles for the best known prior technique [5,8]. The solution is efficient and takes only a few seconds to perform. We are encouraged by these preliminary results and believe Octant provides a general, practical, and principled approach for the geolocalization of Internet hosts.

2 FRAMEWORK

The goal of the Octant framework is to compute a region β_i that comprises the set of points on the surface of the globe where node i might be located. This *estimated location region* β_i is computed based on constraints $\gamma_0 \ldots \gamma_n$. A constraint γ is a region on the globe in which the target node is believed to reside, along with an associated weight that captures the strength of that belief.

Constraints are obtained via network measurements from a set of nodes, called *landmarks*, whose physical locations are at least partially known. Every landmark node L_j has an associated estimated location region β_{L_j}, whose size captures the amount of error in the position estimate for the landmark. We call a node a *primary landmark* if its position estimate was created via some exogenous mechanism, such as a GPS measurement or by mapping a street address to global coordinates. We call a node a *secondary landmark* if its position estimate was computed by Octant itself. In such cases, β_{L_j} is the result of executing Octant with the secondary landmark L_j as the target node.

Octant enables landmarks to introduce constraints about the location of a target node based either on *positive* or *negative* information. A positive constraint is of the form "node A is within x miles of Landmark L_1," whereas a negative constraint is a statement of the form "node A is further than y miles from Landmark L_1."

In the simple case where the location of a primary landmark is known with pinpoint accuracy, a positive constraint with distance d defines a disk with radius d centered around the landmark in which the node must reside. A negative constraint with distance d' defines the complement, namely, all points on the globe that are not within the disk with radius d'. When the source landmark is a primary whose position is known accurately, such constraints define an annulus.

For a secondary landmark k whose position estimate is β_k, a positive constraint with distance d defines a region that consists of the union of all circles of radius d at all points inside β_k (formally, $\gamma = \bigcup_{(x.y)\in\beta_k} c(x,y,d)$ where $c(x,y,d)$ is the disk with radius d centered at (x,y)). In contrast, a negative constraint rules out the possibility that the target is located at those points that are within distance d regardless of where the landmark might be within β_k (formally, $\gamma = \bigcap_{(x,y)\in\beta_k} c(x,y,d)$). Octant's representation of regions using Bezier curves enables these operations to be performed efficiently via transformations only on the endpoints of Bezier segments.

Given a set Ω of positive constraints and a set Φ of negative constraints on the position of a target node i, the estimated location region for the target is given by:

$$\beta_i = \bigcap_{X_i\in\Omega} X_i \setminus \bigcup_{X_i\in\Phi} X_i.$$

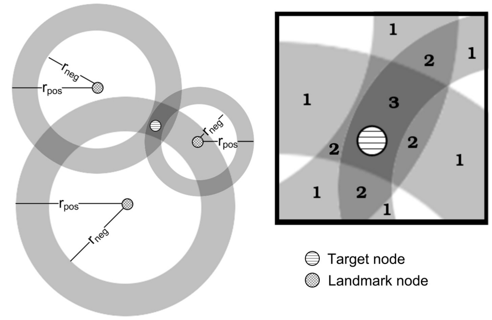

Figure 1: Octant computes an estimated location region for a target node by combining positive and negative information available through latency measurements. The resulting location estimate comprises non-convex, potentially disjoint regions separated by weight.

This equation is precise and lends itself to an efficient geometric solution. Figure 1 illustrates how Octant combines constraints to yield an estimated location region for a target. In this general formulation, the solution is discrete; a point is either part of the solution space or it is not. A discrete solution strategy leads to a brittle system, as a single erroneous constraint will collapse the estimated location region down to the empty set. We show later optimizations that enable the Octant framework to be applied to noisy and conflicting measurements on the Internet.

If latencies on the Internet were directly proportional to distances in the real world, the geolocalization problem would be greatly simplified. In the following sections, we present various techniques for extracting accurate constraints from network-level measurements.

2.1 MAPPING LATENCIES TO DISTANCES

The network latency between a target and a landmark physically bounds their maximum geographical distance. A round-trip latency measurement of d milliseconds between a landmark and a target can be translated into a distance constraint using the propagation delay of light in fiber, approximately $\frac{2}{3}$ the speed of light. This yields a conservative positive constraint on node locations that can then be solved using the Octant framework to yield a sound estimated position for the target. In practice, however, such constraints are so loose that they lead to very low precision.

Yet the correlation between latency measurements and real-world distances is typically better and tighter than constraints based on the speed of light. Figure 2 plots the network latency against physical distance from a primary landmark (planetlab1.cs.rochester.edu) to all other primary landmarks in our study. The figure makes clear the loose correlation between physical distance and illustrates how overly conservative the speed of light bounds can be. In addition, the empty region to the lower right

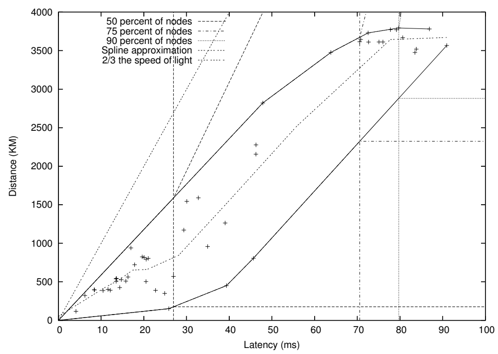

Figure 2: The latency-to-distance plot of peer landmarks for the planetlab1.cs.rochester.edu landmark. The convex hull around the data-points serves as the positive and negative constraints for the node.

suggests that few links are significantly congested; nodes that are physically close are typically reachable in a short amount of time. This presents an opportunity for a system wishing to aggressively extract constraints at the risk of occasionally making overly aggressive claims, to both tighten the bounds on positive constraints and to introduce negative constraints.

Octant calibrates each landmark periodically to determine the correlation between network measurements performed from that landmark and real-world distances. The goal of the calibration step is to compute two bounds $R_L(d)$ and $r_L(d)$ for each landmark L and latency measurement d such that a node i whose ping time is d will be between $r_L(d) \leq ||loc(L) - loc(i)|| \leq R_L(d)$. This permits Octant to extract a positive and a negative constraint for each measurement made from each landmark.

Octant uses a principled approach to pick R and r conservatively such that it can extract bounds that are both tight and likely to be correct. Each landmark periodically pings all other landmarks in the system, creating a correlation table much like Figure 2. It then determines the convex hull around the points on the graph. Functions R_L and r_L correspond to the upper and lower facets of the convex hull.

In practice, this approach yields good results when there are sufficient landmarks that inter-landmark measurements approximate landmark-to-target measurements. In cases where there are just insufficient landmarks to draw statistically valid conclusions, Octant introduces a cutoffs at latency ρ, such that a tunable percentile of landmarks lie to the left of ρ, and discards the part of the convex hull that lies to the right of ρ. Octant then uses $r_L(x) = r_L(\rho), \forall x \geq \rho$, and $R_L(x) = m(x - \rho) + R_L(\rho), m = (y_z - R_L(\rho))/(x_z - \rho)$, where a fictitious sentinel datapoint z, placed far away, provides a smooth transition from the aggressive estimates on the convex hull towards the conservative constraints based on the limits imposed by the speed of light.

2.2 QUEUING DELAYS

Mapping latencies to distances is further complicated by queuing delays introduced by routers and end hosts. Octant uses a fast, low-overhead, end-to-end approach for capturing the minimum queuing delay seen on measurements from a given host in a single, simple metric.

This approach is similar to the height concept in Vivaldi [4] in that it represents the inelastic component of end-to-end latency measurements. However, our derivation is different, and simpler, because our heights capture just the minimum queuing delay.

Octant derives heights and queuing delay estimates from pair-wise latency measurements between landmarks. Primary landmarks, say a, b, c, measure their latencies, denoted $[a, b]$, $[a, c]$, $[b, c]$. Since the positions of primary landmarks are known, the great circle distances between the landmarks can be computed, which yield corresponding estimates of transmission delay, denoted (a, b), (a, c), (b, c). This provides an estimate of the queuing delay between any two landmarks; for instance, the queuing delay between landmarks a and b is $[a, b] - (a, b)$[1]. Octant determines how much of the delays can be attributed to each landmark, denoted a', b', c', by solving the following set of equations:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix} = \begin{bmatrix} [a,b] - (a,b) \\ [a,c] - (a,c) \\ [b,c] - (b,c) \end{bmatrix}$$

Similarly, for a target t, Octant can compute t',as well as a estimate of the longitude and latitude, t_{long} and t_{lat}, by solving the following system of equations.

$$\begin{aligned} a' + t' + (a, t) &= [a, t] \\ b' + t' + (b, t) &= [b, t] \\ c' + t' + (c, t) &= [b, t] \end{aligned}$$

where (a, t) can be computed in terms of a_{long}, a_{lat}, t_{long}, t_{lat}. We can then solve for the t', t_{long}, t_{lat} that minimizes the residue. The computed t_{long} and t_{lat} result, similar to the synthetic coordinates assigned by Vivaldi, has relatively high error and is not used in the later stages.

Given the target and landmarks' heights, each landmark can adjust their latency measurements to more accurately approximate the transmission delay component.

2.3 INDIRECT ROUTES

The preceding discussion made the simplifying assumption that route lengths between landmarks and the target

[1] Note that this difference might embody some additional transmission delays stemming from the use of indirect paths. We expand on this in the next section.

are proportional to great circle distances. In practice, policy routing often leads to network paths that differ from great circles. A geolocalization system with a built-in assumption of proportionality would not be able to achieve good accuracy.

The height computation used to isolate queuing delays addresses some, but not all, of the inaccuracies stemming from indirect routes. However, it does not address inaccuracies from the inconsistent or unexpected use of indirect routes. This occurs often enough in practice that accurate geolocalization requires a more targeted and exact mechanism to compensate for its effects.

Octant addresses indirect routes by performing piecewise localization, that is localizing routers on the network path from the landmarks to the target serially, using routers localized on previous steps as secondary landmarks. Localization of routers can be further refined by leveraging the structured way many routers are named. Octant performs a reverse DNS lookup on each router on the path and tries to determines the city in which it resides by using the **undns** [9] tool. This approach yields much better results than using just end-to-end latencies, as routes between routers separated by a single link is largely void of indirect routing.

2.4 HANDLING UNCERTAINTY

A mechanism to handle and filter out erroneous constraints is critical to maintaining high localization accuracy. The core mechanism Octant uses is to assign weights to constraints based on their inherent accuracy.

For latency-based constraints, we have observed that constraints from landmarks that have high latency to the target are less trustworthy than those that are nearby. Octant uses a weight system that decreases exponentially with increasing latency, thereby mitigating the effect of high-latency landmarks when lower latency landmarks are present. When two regions overlap, Octant determines all possible resulting regions via intersections, and assigns the associated weight to each. The final estimated location region is computed by taking the union of all regions, sorted by weight, such that they exceed a desired size threshold. Bad constraints may still impact accuracy if there are no compensating factors, but weights enable Octant to associate a probability measure with regions of space in which a node might lie.

2.5 GEOGRAPHICAL CONSTRAINTS

In addition to constraints extracted from latency measurements, Octant enables any kind of geographical constraint to be integrated into the localization process. In particular, Octant makes it possible to introduce both positive (such as zipcodes from the WHOIS database, zipcodes obtained from other users in the same IP prefix [8]) and negative constraints (such as oceans, deserts, uninhabitable areas) stemming from geography and demographics. In prior work, which does not permit non-convex regions, the removal of such areas typically requires an ad-hoc post-processing step. In contrast, Octant can naturally accommodate such constraints.

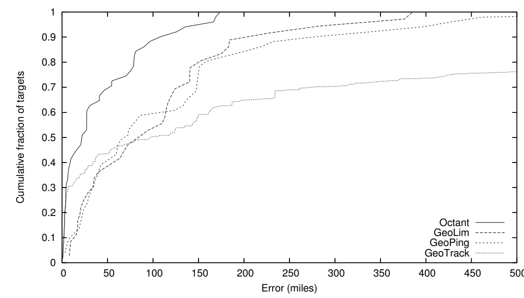

Figure 3: Comparison of the accuracy of different localization techniques. Octant achieves significantly greater accuracy than previous work, yielding point estimates for nodes that are substantially closer to the real positions of the targets.

3 EVALUATION

We evaluated Octant using physical latency data collected from 51 PlanetLab [3] nodes whose real world geographic locations we were able to determine externally. The latency data was collected via 10 time-dispersed round-trip measurements using ICMP ping probes. To evaluate the efficacy of using secondary landmarks, we also collected the full traceroute information between every landmark pair, as well as latency data between the landmarks and intermediate routers. Following [8, 5], nodes serve both as landmarks and targets in our evaluation; of course, the node's own position information is not utilized when it is serving as a target. No two hosts in our evaluation reside in the same institution, which rules out simple yet unrealistic and unscalable solutions to geolocalization that rely on having a nearby landmark for every target. We compare Octant with GeoLim, GeoPing, and GeoTrack, the current state-of-the-art in geolocalization.

Figure 3 shows the accuracy of different geolocalization techniques by plotting the CDF of the distance between the position estimate and the physical location of a node. Octant is significantly more accurate than the other techniques, because it represents regions precisely, extracts tighter constraints from the measurements, and solves the system of constraints without introducing errors in the process. Octant achieves a median error of 22 miles, compared to 89 miles for GeoLim, 68 miles for GeoPing and 97 miles for GeoTrack. Octant's results are significantly better even for the tail of distribution; its worst-case error was 173 miles, in contrast to 385, 1071, and 2709 miles for GeoLim, GeoPing and GeoTrack, respectively.

In a typical deployment, the number of landmarks used

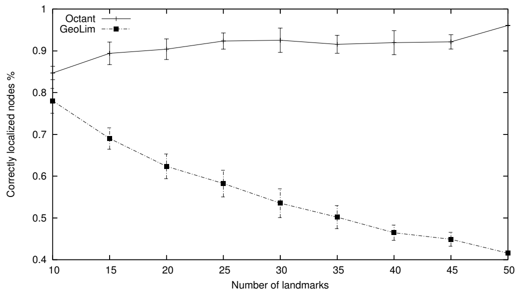

Figure 4: The percentage of targets inside the Octant's location estimate is significantly higher than GeoLim's.

to localize a target is often constrained by physical availability. We evaluate Octant's performance as a function of the number of landmarks used to localize targets, and compare to GeoLim, the only other region-based geolocalization system. Figure 4 shows the number of nodes that were located successfully; that is, their physical locations were inside the estimated location region. The percentage of nodes that successfully localized is quite high for Octant, even with only 10 landmarks. Surprisingly, the accuracy of the GeoLim approach drops as more landmarks are introduced. This behavior is due to overaggressive extraction of constraints in GeoLim; as the number of landmarks increases, the chances that a "bad" node that will introduce an overconstraint grows. The use of weighted combination of constraints enable Octant to avoid this pitfall.

4 RELATED WORK

Past work on mapping nodes to their locations on the globe has focused mostly on using positive information for determining a single point estimate for a node.

IP2Geo [8] proposes three different techniques for geolocalization, called GeoPing, GeoTrack and GeoCluster. GeoPing maps the target node to the landmark node that exhibits the closest latency characteristics, based on a metric for similarity of network signatures [2]. GeoTrack performs a traceroute to a given target, extracts geographical information from the DNS names of routers on the path, and localizes the node to the last router on the path whose position is known. GeoCluster is a database based technique that first breaks the IP address space into clusters that are likely to be geographically co-located, and then assigns a geographical location to each cluster based on IP-to-ZIP mappings from third party databases, such as user registration records.

GeoLim [5] derives the estimated position of a node by measuring the network latency to the target from a set of landmarks, extracts upper bounds on position based on inter-landmark distance to latency ratios, and locates the node in the region formed by the intersection of these fixes to established landmarks.

Services such as NetGeo [7] and IP2LL [1] geolocalize an IP address using the locations recorded in the WHOIS database for the corresponding IP address block. The granularity of such a scheme is very coarse for large IP address blocks that contain geographically diverse nodes.

Localization has been studied extensively in wireless systems. The most comprehensive work on localization in wireless networks is Sextant [6]. We share with Sextant the basic insight for accommodating both positive and negative constraints and enabling constraints to be used by landmarks whose positions are not known definitively. Octant differs substantially from Sextant in the various mechanisms it uses to translate Internet measurements to constraints.

5 SUMMARY

Octant provides a general and comprehensive geolocalization framework that can accommodate any set of constraints, extract aggressive constraints, and solve the resulting system accurately. The system is practical, with solution times under a few seconds including the time for network measurements, and has already been deployed. Octant opens up the possibility of enabling network operators to determine node location on-demand without resorting to unreliable and inaccurate IP-to-ZIP databases. We hope that accurate data on node position will be used for customized content delivery, network management and network diagnosis, without compromising user privacy.

References

[1] IP to Latitude/Longitude Server, University of Illinois. http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll.

[2] P. Bahl and V. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, Tel Aviv, Israel, March 2000.

[3] A. Bavier, M. Bowman, B. Chun, D. Culler, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating System Support for Planetary-Scale Network Services. In *Networked Systems Design and Implementation*, San Francisco, CA, March 2004.

[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A Decentralized Network Coordinate System. In *SIGCOMM*, Portland, OR, August 2004.

[5] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-Based Geolocation of Internet Hosts. In *Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004.

[6] S. Guha, R. Murty, and E. Sirer. Sextant: A unified framework for node and event localization in sensor networks. In *Mobihoc*, Urbana-Champaign, IL, May 2005.

[7] D. Moore, R. Periakaruppan, and J. Donohoe. Where in the World is netgeo.caida.org? In *INET2000 Poster*, Yokohama, Japan, July 2000.

[8] V. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for internet hosts. In *SIGCOMM*, San Diego, CA, August 2001.

[9] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *SIGCOMM*, Pittsburgh, PA, August 2002.