Alidade: IP Geolocation without Active Probing

Technical Report CS-TR-2015.001
Department of Computer Science, Duke University
January 2015
(revised April 2015)

Balakrishnan Chandrasekaran[*], Mingru Bai[*], Michael Schoenfield[*], Arthur Berger[†], Nicole Caruso[‡]
George Economou[†], Stephen Gilliss[†], Bruce Maggs[*†], Kyle Moses[°], David Duff[†]
Keung-Chi Ng[†], Emin Gün Sirer[‡], Richard Weber[†], Bernard Wong[∓]

[*]Duke University, [†]Akamai Technologies, [‡]Cornell University, [°]U.S. Military Academy, [∓]University of Waterloo

ABSTRACT

Geolocation systems generally fall into two categories. Commercial systems provide precomputed address-to-location mappings for all IP addresses. We refer to such systems as geolocation databases. Upon presenting a geolocation database with a target IP address, a location estimate is provided immediately. Almost all systems reported in the academic literature, on the other hand, employ active measurements, issuing probes to a target after it has been specified, but before estimating the location of the target. These systems use constraints derived from the measurements to improve the accuracy of their predictions. Both approaches have their advantages. The active measurement approach may be more accurate, while the geolocation database approach is not intrusive and can answer queries quickly, even when off-line. This paper presents Alidade, a geolocation database system that makes extensive use of available network measurement data, but does not issue any probes of its own, either before or after a target is presented. Like other geolocation databases, Alidade precomputes location estimates for all of IP space. Indeed, using the available constraints, Alidade computes a joint solution for all addresses. We demonstrate that Alidade is competitive with the best commercial systems – on their own terms – using six different ground-truth data sets. Alidade also provides stronger guarantees of correctness, and each of Alidade's predictions consists of a geographical region in addition to a representative point.

1. INTRODUCTION

This paper introduces a new geolocation system called *Alidade*. Geolocation systems accept queries of the form, "Where is 128.2.205.42?" and then provide predictions, such as, "128.2.205.42 is in Pittsburgh, Pennsylvania." The geolocation problem has been studied extensively by the networking research community, and we forgo the customary explanation of its importance. Alidade, however, is fundamentally different from previous systems described in the academic literature because it computes predictions for the entire IP address space and does not issue any measurement probes of its own, either before or after it is presented with queries. Instead, Alidade fuses available data sets of various types, attempting to resolve conflicts in the data and to find mutually compatible solutions for all addresses.

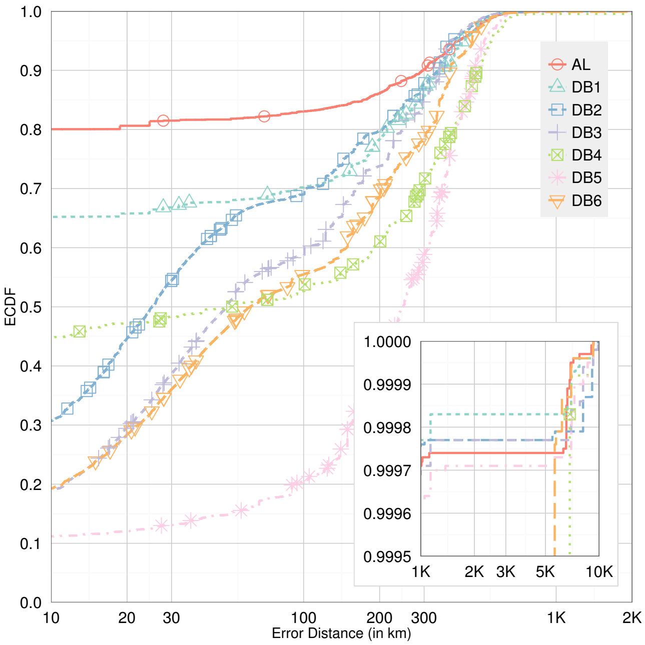

Figure 1: Comparison of Alidade's geolocation accuracy with six commercial geolocation databases

Commercial geolocation databases also provide precom-

This research was supported in part by AFRL Grants FA8750-11-1-0262 and FA8750-10-2-0193.

puted answers for all IP addresses. Like Alidade, the commercial products do not issue any probes when presented with geolocation queries. Alidade competes head-to-head with these databases, and, as Figure 1 shows, outperforms even the best of them on a large ground-truth data set provided by a Tier-1 ISP. We compare and contrast Alidade's geolocation accuracy with that of six other geolocation database systems: *EdgeScape (ES)*, *MaxMind GeoCity (MM)*, *MaxMind GeoCity2 Lite (MML)*, *DB-IP (DBIP)*, *IP2Location (IP2L)*, and *IPligence (IPLG)*. EdgeScape is a leading commercial geolocation database and is an Akamai offering. As part of our collaboration with Akamai, we had full access to all of the data that EdgeScape uses and full knowledge of the algorithms used by EdgeScape, and yet have found it a great challenge to improve significantly on its results.

As part of the evaluation, the systems were presented with 100,000 targets sampled uniformly at random from the ground-truth data set. Figure 1 shows the error distance (in km) on a log-scale along the x-axis and the Empirical Cumulative Distribution Function (ECDF) of these errors along the y-axis; we define error distance as the distance between the point-based prediction made for a target address and its ground-truth location. Alidade outperforms the other six systems with 79% of its targets geolocated to within a 10km error. Because the exact methods used to compile the commercial databases are proprietary, we do not know for certain why Alidade is more accurate.

Our analysis of Alidade includes a breakdown of how much each type of data aids in making accurate predictions. Not surprisingly, no single source of data suffices to make good predictions. The data sets ingested by Alidade include latency and path measurements collected for other purposes, e.g., traceroute data from iPlane [21] and CAIDA's Archipelago (Ark) measurement infrastructure [4], and client-server round-trip times measured by a Content Delivery Network (CDN). Alidade also relies on a tool called *HostParser* that translates domain names into geographical locations, much as the Undns tool [28] does. To provide coverage over the entire IP address space, Alidade leverages data from the Internet registries too. The extent to which the registry entry for an address is trusted is mitigated by the position of the corresponding Autonomous System (AS) in the AS hierarchy produced by CAIDA [5].

At its core, Alidade is a constraint-based *passive* geolocation system, inspired by Octant [31], but able to incorporate a wider variety of non-measurement data sources. Alidade uses latency measurements only when they are issued from hosts with known geographical locations, e.g., PlanetLab nodes. We call these hosts and/or their IP addresses *landmarks.* Alidade's estimate of the location of an address with an unknown location, which we call a *target,* is represented as a *polygonal* region on the surface of the Earth that should (if the prediction is correct) contain the address. The predictions made by commercial geolocation systems, in contrast, generally consist of a single latitude-longitude point or the name of a city or country. To facilitate a comparison with these systems, Alidade selects a single point to represent the polygon region. Although sophisticated techniques based on population density maps could be used to pick the representative point, at present Alidade just uses a set of heuristics that select the center of some city contained in the answer region.

Figure 2 shows an example of an answer region computed by Alidade. The region bounded by the dark green line represents the area resulting from intersecting constraints derived from latency measurements. In this example the intersection happens to be a circular region. The polygon in blue is a country-level hint (Germany) inferred from one of the Internet registries. Since the registry data does not conflict with the contraints derived from the measurements, Alidade uses it to further refine its prediction. In this example, Alidade has also identified a city-level hint (Kaiserslautern, a district in the Rhineland-Palatinate state of Germany) by examining the names of the routers on a traceroute path to the target. The city-level hint is indicated in the figure by the tiny red polygon inside the larger blue one. Ultimately Alidade pins the target in this demonstration to Kaiserslautern, which is consistent with the ground truth location of the target.

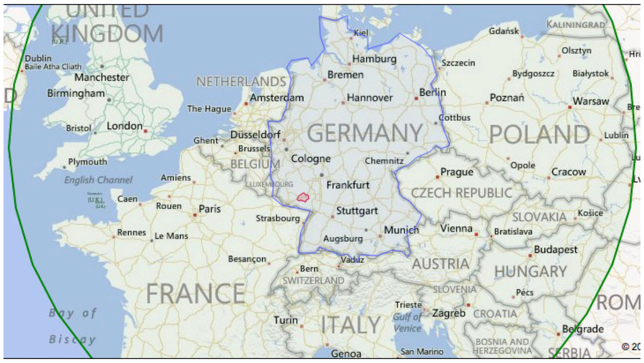

Figure 2: Example of a prediction made by Alidade for a target.

To process large volumes of data, Alidade is structured as a map-reduce (Hadoop) application. (Indeed, we started by porting Octant to Hadoop.) We conducted our experiments using a cluster of 40 8-core servers, each with 32GB of RAM. Each component of Alidade exhibits "embarrassing parallelism" and is implemented as a map-reduce job. In a later section we provide a breakdown of where the Alidade application spends most of its time, e.g., in "preprocessing" measurement data.

2. RELATED WORK

Past work on IP geolocation can be loosely categorized into *active* approaches that perform on-demand network measurements to derive constraints on a target's geographic location, and *passive* approaches that rely only on previously collected information to geolocate a target. Both approaches have advantages and disadvantages. Active approaches may be more accurate, but predictions may not be available until new measurements have been taken. Passive approaches can precompute predictions and hence answer queries immediately, without even requiring network access at query time. Importantly, passive approaches are also unobtrusive, and do not risk alerting or annoying the target of a prediction. But passive approaches may not have the target-specific measurement data that would enable better accuracy.

Alidade takes a passive geolocation approach, but Alidade does not rely exclusively on coarse-grained and potentially error-prone data, such as the WHOIS database and hostname-to-location hints. Instead, Alidade filters the hints provided by these data sets by applying constraints derived from large volumes of passively collected network measurements.

In the following sections we examine both active and passive approaches, noting where Alidade borrows techniques.

2.1 Active Approaches

Much of the prior work in geolocating IP addresses relies on on-demand network measurements. *IP2Geo* [26] is an early IP geolocation system that introduces two active IP geolocation techniques. The first technique is *GeoPing*, which requires a deployment of landmarks of known geographic locations that can perform all-pairs latency measurements. To predict the location a target, all landmarks probe the target. *GeoPing* then selects the landmark that has the most similar latency profile (the set of latency measurements from other landmarks) to the user-specified target. It then uses the landmark's location as the prediction for the target. Although this technique is simple and easy to deploy, the location of a target cannot be accurately predicted unless there is a landmark nearby and that landmark has a similar latency profile. At present, Alidade doesn't compile latency profiles or compare the latency profiles of targets and landmarks. The second technique is *GeoTrack*, which performs traceroutes from landmarks to the target to discover routers on the traceroute paths whose DNS names can be interpreted geographically. From this set of routers, GeoTrack locates the target at the closest router's location, where distance is determined in terms of estimated network latency. Alidade's "extrapolator" applies a variation of this technique. By relying only on this relatively incomplete data source, however, GeoTrack's geolocation accuracy is inconsistent.

In contrast to locating the target at the closest landmark or router, Constraint-Based Geolocation (CBG) [14] determines the location of a target by creating circles on the surface of the earth around each landmark, where each circle represents a constraint that bounds the possible location of the target. The size of each circle is a function of the latency between the landmark and target. CBG combines constraints by intersecting the circles, and selects the middle of the intersection as its best estimate of the target's location. One risk in taking this approach is that a single corrupt measurement can lead to an empty intersection. At its core, Alidade is a CBG approach.

Octant [31] builds on CBG by providing a general framework that can combine both positive and negative constraints, that is, information on where the target is likely and unlikely to be, respectively. To handle uncertain or error-prone data sources, Octant combines constraints using a weight-based mechanism that can limit the impact of erroneous measurements. Alidade builds on the Octant framework. In order to process large volumes of measurement data and to geolocate all of the IP address space, Alidade restructures the framework into a parallel Hadoop application so that more memory and compute cycles can be applied.

Topology-Based Geolocation (TBG) [18] uses traceroutes from the landmarks to the target to discover the routers along the network paths and determine inter-router latencies. With this data, TBG performs a global optimization to find a physical placement of the routers and the target that minimizes inconsistencies with the network latencies. By attempting to globally optimize the placement of both the routers and the target, TBG is more sensitive to measurement errors, such as inflated latencies, than constraint-based solutions, where errors tend to be more localized. To some extent Alidade applies this approach too. In particular, Alidade uses all available estimated latencies between pairs of addresses (landmarks, routers, and end hosts) to jointly predict the locations of the routers and end hosts.

Several systems [11, 32, 2, 19] have applied statistical approaches to construct landmark-specific functions that map measured latencies to geographical distances. These systems generally have significant computational requirements, and are currently unable to make use of non-latency-based constraints. *Posit* [10] presents a more recent statistical approach that, while still requiring active measurements, is able to significantly reduce the required number of on-demand probes by precomputing a statistical embedding. At present, Alidade does not construct a sophisticated model of the relationship between latency and distance. Instead, Alidade uniformly assumes that datagrams travel at two-thirds the speed of light, which is very close to the speed of light in optical fiber. Hence, in converting latency to distance, Alidade does not model circuitous fiber paths, nor does it model queuing delays or any other sorts of delays. The resulting constraints tend to be loose, but they are also hard. In particular, provided that no measurements are corrupt and no faster-than-fiber technologies, such as microwave transmission, are employed, the intersection of a set of constraints derived by Alidade from direct latency measurements must contain the actual location of the target. Other work has suggested that if latency is to be converted to distance by a simple multiplicative factor, four-ninths the speed of light might be used. The smaller constant leads to smaller intersection areas, but these areas might be empty or might not contain the target.

Guo et al. [15] propose mining physical addresses displayed on publicly accessible Web sites that are hosted by Web servers with IP addresses in the same prefix as the target address, and using these physical addresses as hints to improve geolocation accuracy and as sources of ground truth to support evaluations. Caruso [6] (as part of the Alidade project) and Wang et al. [29] extend this approach by combining the mined information with latency measurements to offer finer-grained geolocation results. Although these systems produce accurate results in certain experiments, it is difficult to ascertain their actual effectiveness in general. First, it is tricky to determine when an organization is hosting its own Web site. Furthermore, even when an organization does host its own site, for the technique to work the site must list a physical address that is close to that of the hosting location. In previous experiments the best results were obtained when the set of geolocation targets were biased towards belonging to organizations that typically host their own Web servers and publish physical address information on their web pages, e.g., in one experiment reported in [29], university Web servers hosting Web pages listing campus addresses were used as landmarks and PlanetLab nodes were used as targets. Nevertheless, scraped address information from locally-hosted Web sites is a rich source of geographic data, and Alidade includes this information as one of its many data sources.

Gill et al. [13] propose two broad classes of attacks on active measurement-based geolocation approaches. The first misleads geolocation systems by injecting delays to latency probes from specific landmarks at the target, thereby altering the geolocation result by moving the centroid of the constraint intersection in a CBG-based approach. The second targets topology-aware geolocation approaches by altering inter-router latencies in traceroutes, which enables powerful adversaries to place geolocation targets at arbitrary locations. Alidade does not attempt to detect possible adversaries. Unlike active approaches, however, where latency probes can often be easily identified, Alidade also uses a large body of passively collected measurements that piggyback real user TCP connection requests and replies. Adversaries must therefore delay legitimate TCP traffic rather than just latency probes in order to distort much of Alidade's input data.

There has also been considerable work on using active measurements to assign artificial coordinates to Internet nodes. The latency between a pair of nodes is then estimated by computing the distance between the two nodes in the artificial coordinate space. *GNP* [25] is a pioneering work in this area. *GNP* embeds nodes into a low-dimensional Euclidean space, where the distance between two nodes in the space approximates the network latency between the nodes. There is no guarantee that the artificial coordinates map in any natural way to the true physical locations of the nodes on the surface of the Earth, however, nor is this a goal of GNP. Building on GNP, *Vivaldi* [8] introduces a decentralized network embedding approach that obviates the need for fixed landmarks. *Meridian* [30] introduces an overlay routing approach to solve network positioning problems without needing to perform an explicit network embedding. This enables Meridian to avoid intrinsic network embedding errors. Alidade (whose goal is not to predict latencies) does not have much in common with these approaches.

2.2 Passive Approaches

Although active geolocation approaches can be highly accurate, their dependence on performing on-demand network measurements make them unsuitable for many location-aware applications. Most commercial geolocation systems, such as *MaxMind GeoCity* [22], *EdgeScape* [1], *IPInfoDB* [17], and *HostIP.Info* [24] have instead adopted passive approaches, where they offer their users a pre-computed IP-to-location database that can identify a target's location without additional network access. Unfortunately, the exact methodology for creating these databases are generally proprietary; only the expected accuracy of these databases are typically published. However, the common understanding is that these databases rely on a combination of domain registry information, ISP provided data, host name hints, latency measurements, and other heuristics. Alidade relies on many of the same sources, except that the ISP-supplied ground-truth geolocation data (from one Tier-1 ISP) is used only for evaluation purposes and not as an input to Alidade.

Poese et al. [27] performs an analysis of the accuracy of commericial geolocation databases. They report that while geolocation databases are extremely accurate at the country level, they perform poorly at the city level. Note that Poese et al. did not analyze EdgeScape (or Alidade).

In addition to GeoPing and GeoTrack, IP2Geo [26] also introduces *GeoCluster*, a passive approach that partitions the IP address space into geographically co-located clusters. GeoCluster then assigns each cluster to a geographic location based on the geographic information extracted from user registration and usage databases. The effectiveness of this approach is largely limited by the availability of such databases, the geographic coverage of the users in the databases, and the accuracy and freshness of the self-reported user location information. At present, no such data is available to us, but if it were, it could be used as an input to Alidade.

3. SYSTEM DESIGN

We built Alidade to assimilate data that is large in volume and rich in diversity. Alidade consists of many components, each of which is designed as a map-reduce job. The components are composed into a pipeline, with the intermediate results persisted using *HDFS* and *HBase*. Figure 3 shows a high-level overview of the system; Alidade's components are indicated by the red blocks and the ordering of these in the illustration, from left to right, corresponds with their positions in the system's pipeline. In the rest of the document, the term Alidade refers to this pipeline or workflow in its entirety.

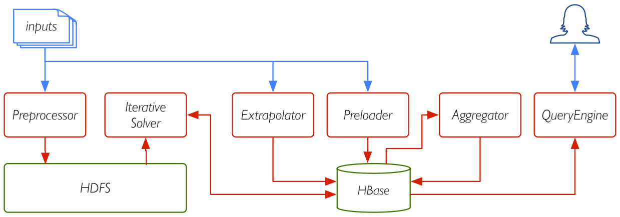

Figure 3: Alidade: System Overview

3.1 Preprocessor

To exploit measurement and non-measurement data sets already available from various projects, and consequently, avoiding active probing within Alidade, the system is designed to accept a wide variety of inputs. Typically, the input comprises of ping measurements, traceroute data, HostParser answers, and Internet registry information. The *preprocessor* processes the input measurement data and converts them into a standardized internal format. The conversion to an internal format, to some extent, allows components further down the pipeline to be oblivious to the heterogeneity in input.

The I/O bound preprocessing phase represents the most time-consuming component of Alidade. In the experiments described in this paper, eight hours was typical. This map-reduce job, however, needs to be run only once against any input data. In addition to the parsing and transformation functions, the preprocessor summarizes the distribution of measurements (latencies) between a pair of landmark and target, by a single value, which we use as an approximation of the actual latency between the pair. The preprocessor does not necessarily pick the smallest latency from the distribution of observed values. Typically, the preprocessor chooses the median latency; the choice, however, can shift to the mean or the minimum, depending on the distribution of observed round-trip times.

Our concerns about using the minimum measured latency were fueled by an analysis of measurements recorded by the *iPlane* [21] project on the PlanetLab platform over several years. Using the ground truth locations reported for the PlanetLab nodes, we computed the minimum latency possible between each pair of nodes. For each pair, the minimum (or threshold) latency value is computed by taking the true distance between the two nodes over the surface of the Earth and dividing by two-thirds the speed of light, which is the speed of light in optical fiber. We then scanned the data for all traces in which the recorded latency is smaller than the threshold by at least 10ms. Figure 8 provides a CDF of the iPlane traceroute measurements with such speed-of-light violations for 2010 and 2011. In 2010 there were 95,188 such measurements, in 2011 there were 4,031. The x-axis, in log-scale, shows the magnitude of error, i.e., the value by which the observed latency in the traceroute is smaller than the computed threshold. Errors in the reported ground truth locations of landmarks are a known cause for measurement inconsistencies, and we discovered several errors in the reported locations of PlanetLab nodes. But most errors could not be explained by bad reported locations. For example, the vast majority of the 2010 errors originated at Peking University, while the vast majority of the 2011 errors originated at USC ISI, both of which report their locations correctly. In summary, the plot provides a warning against simply using the minimum observed latency.

Preprocessor's outputs are persisted in HDFS as serialized binary objects. Output consists of an observation for every landmark and target pair observed in the input data sets. Observations are categorized into two classes: *direct* or *indirect*. Direct observations are latency measurements reported directly or explicitly by latency-based measurement tools, viz., ping or traceroute. Indirect observations are inferred latencies between intermediate hops on the path taken by a packet from a source to destination; Figure 4 shows direct latencies in red, and indirect in blue, on a path revealed by a tool like traceroute. The indirect observation BC, in the illustration, is computed by taking the difference between direct observations AC and AB. Path asymmetries and queuing delays along the path often introduce errors in indirect observations, including negative latencies!

3.2 Iterative Solver

The *iterative solver* reads the observations output by the preprocessor and combines them with non-measurement data, viz., HostParser answers. The solver geolocates all targets observed in the input, in parallel, in a map-reduce job. Location estimates are regions formed by intersecting two or more constraints, and improved with non-measurement data, if any. The estimates are persisted in HBase and refined further in each iteration. Each iteration executes the same sequence of functions – derive constraints from observations, solve the constraints and combine the solution with non-measurement data. Iterations after the first, also read as inputs the answers generated for targets in the previous iteration. We typically iterate the solver three times, after which the gains from iterations seem to increase only marginally. Direct measurement data take the highest precedence amongst data sources used in Alidade; non-measurement data that do not overlap with the direct measurements are therefore discarded, without exception.

To derive constraints from latencies, Alidade multiplies them with two-thirds the speed of light. The constraint is an N-sided polygon, with N typically set to 32. The center of the constraint is the location of the source of the observation. Generating a constraint from a direct observation is straightforward, since the source is a landmark; the location of a landmark is known *a priori*. Assume, for instance, that a network probe takes a path from landmark A to target C via an intermediate hop B, and another that starts at landmark

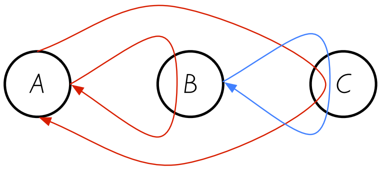

Figure 4: Observation Types

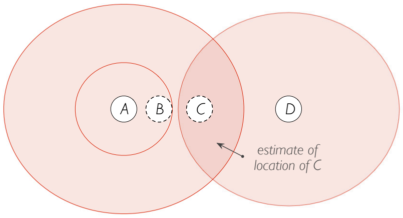

Figure 5: Direct Constraint

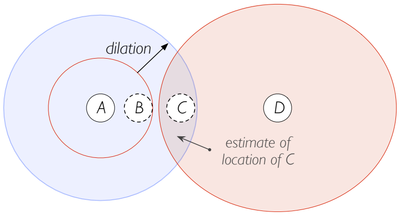

Figure 6: Indirect Constraint

D and reaches C, in a single hop. The constraints for targets B and C using the direct measurements from A would be polygons centered at A and sized proportional to the latencies observed, as shown in Figure 5. The illustration also shows the intersection of two constraints, one from A and the other from D, generating a location estimate for C; note that although the illustration makes use of circles for constraints, Alidade represents them as polygons.

An indirect observation cannot be used in the first iteration, because the location of the source of observation has not been estimated until after the first iteration. For later iterations, the location of the source of an indirect observation is available, but unlike the location of a landmark, it takes the form of a region bounded by a polygon. To generate constraints to a target from an indirect observation, the polygonal region is dilated by a distance proportional to the indirect latency. The intersection of indirect and direct constraints results in a smaller area, refining the location estimate from the previous iteration. Figure 6 illustrates the derivation and use of indirect measurements. Since the source of an indirect observation is also a target, whose location changes after the first iteration, each iteration, theoretically, refines the estimates for all targets with indirect observations. Octant [31] was the first system to demonstrate the technique of exploiting indirect measurements to improve geolocation accuracy.

3.3 Extrapolator

Extrapolator attempts to guess the city location of a target by looking at the names of the routers on a traceroute path to the target. It applies the heuristic that if two hops on a traceroute are close to one another (have few hops in between and a short estimated latency), then they are likely to be located in the same city. Hence, a hint for the location of a target can be determined by scanning a traceroute to the target backwards to find the first router with a location hint. Assuming that this router is close to the target, extrapolator copies the router's hint to creates a location hint for each of the hops on the trace from the router to the target. In our analysis of traceroute data, we found that location hints from routers farther than eight hops from a target or with an indirect latency greater than 60ms were most likely erroneous. Alidade guards against erroneous extrapolator hints for a target by checking them for consistency with constraints on the target's location derived from direct measurements. Extrapolator is one of the more time consuming stages in the pipeline. In our experiments this stage typically took an hour and a half to two hours to complete.

3.4 Preloader

The *preloader* map-reduce job updates the results database with HostParser answers for targets that have no location estimates at this point in the pipeline. The input data for preloader consists of the complete HostParser data set, containing answers for approximately 700 million IP addresses. Less than half of these answers, approximately 211 million, are at city level. Targets with city-level answers from host parser are treated as equivalent to having ground truth, unless they have contradicting latency measurements. When measurement data is available for a target, they always take precedence over any non-measurement data that exist for the same. Preloader is the first component in the pipeline that is concerned with geolocating targets with no measurement data. The answers populated by the preloader, prime the system for generating better answers for the entire routable IP space. Preloader is also fairly time consuming, averaging about forty-five minutes to one hour in our experiments.

3.5 Aggregator

Aggregator summarizes the location predictions that have been made in previous stages of the pipeline for IP addresses within a prefix. It applies the heuristic that addresses in the same prefix are likely to be close to each other geographically, and uses these summaries to make predictions for other addresses in the prefix that lack predictions from earlier stages in the pipeline. Rather than examining every possible prefix (of every length), aggregator builds a prefix tree containing the minimum number of prefixes needed to capture the addresses for which predictions have been made in earlier stages. In particular, the tree is pruned so that it does not contain any two prefixes that contain the exact same set addresses for which predictions have been made earlier. The answers for the addresses within a prefix are either overlapping or non-overlapping, resulting in an *aggregate intersection* or *aggregate union*, respectively.

For a target without measurement data, the longest prefix containing the target provides an initial location estimate. The initial estimate, can be revised later depending on what other non-measurement data is available for the target. For instance, availability of a city-level HostParser takes priority over any aggregate. Aggregates are also applied to targets with measurement data. Measurements available for a target are used as a filter to discard inconsistent answers in the aggregate. Put in a different way, aggregates are recomputed for targets based on the initial measurement-based estimates available for them. Section 4 includes results on improvement in geolocation accuracy from use of aggregates.

3.6 Query Engine

The *query engine* provides an interface through which location estimates generated by the system can be queried and output. Queries can retrieve answers for either specific targets or a subnet. For answers with measurement data, the querying process is a straightforward lookup of results computed for that target from the database. The querying engine also uses aggregates containing the target to further improve this initial location estimate. Alidade assigns the highest precedence to measurement data, and the initial estimates computed using measurement data can be used to effectively trim inconsistent answers in an aggregate; if the set of answers in a prefix that's consistent with the measurement data available for the target have more specific location hints, say at city-level, then the combination might improve the geolocation accuracy.

The querying process for a target with no measurement data is more involved. The initial estimate for such a target is an aggregate that contains the target. However, the query engine can override this estimate with non-measurement data from HostParser or registry, both of which have higher precedence compared to the aggregate; in such scenarios, the non-measurement data becomes the new initial estimate. This initial estimate is refined further by looking for answers from larger subnets (shorter prefixes), if necessary. It is possible, for instance, that the shortest subnet (longest prefix) used to generate the initial estimate does not have a city-level answer. In such situations, the query engine scans for larger subnets, provided that such aggregates contain fine-grained answers.

3.7 Exploiting Registry Data

Various stages in the Alidade pipeline make use of hints derived from the Internet registries. One difficulty with exploiting registry data is that the operator of a network that spans a large geographical area may list a single physical address, which should not be trusted equally to a registry entry for a small regional network. Alidade uses a network's position in the Autonomous System (AS) hierarchy to augment decisions to ignore a registry entry, or trust it for a country or city-level hint.

AS hierarchy information is taken from CAIDA's AS Ranking project [5] [20] and is combined with Alidade's Internet Registry data. Internet Registry entries are mapped to BGP ASes by analyzing AS Path information from routing tables. Consistent paths from multiple landmarks indicates a clear origin AS. Entries with ambiguous origin are often transit networks and are generally ignored by the registry module since they have unclear localization. Alidade's also use a network's prefix size for hint decisions in addition to it's AS rank. These metrics allow us to identify low tier and stub networks advertising small prefixes, which have consistently strong [12] localization.

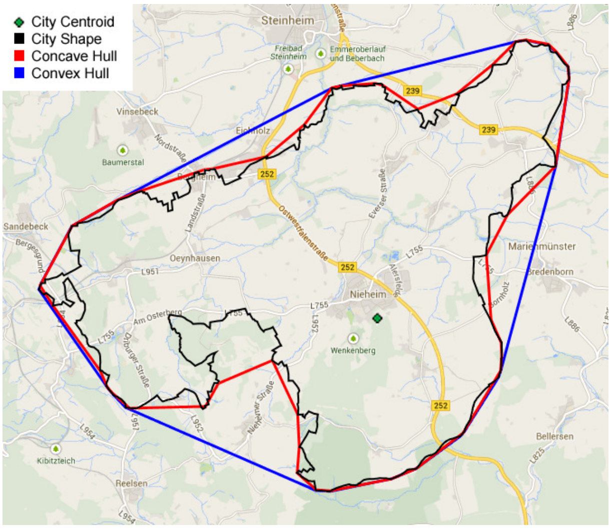

Figure 7: Shape Simplification

Internet Registry data may be checked for any IP address referenced by Alidade. For a large job this could result in hundreds of millions of queries to Alidade's Internet Registry datastore. Minimizing the latency of registry queries can therefore have a significant impact on Alidade's runtime. In order to maximize efficiency, Alidade instantiates a PATRICIA trie [23] within each task JVM in the cluster. PATRICIA tries take advantage of the hierarchical nature of IP address space improving latency and memory utilization. To ensure consistency across the cluster, Registry data is delivered using Hadoop's DistributedCache system.

3.8 Matching City Names to Shapes

A non-trivial problem encountered when exploiting data sources containing city name hints is to convert these names into shapes. Alidade's registry database and HostParser table contain more than 100,000 locations (and associated coordinates) from every country/region on Earth. Location names are listed as compact ASCII strings consisting of ISO two digit country and region codes followed by the city/location name. For example, DE-NI-OSNABRUCK is Osnabrück in the German state of Lower Saxony (Niedersachsen). Mapping these location names to representative shapes is conceptually simple, but nuanced in execution.

First, we compile our database from multiple open sources including postal and census bureaus around the globe. The two primary sources are TIGER/Line 2013 (United States) [7] and GADM (worldwide) [16]. Once a source is loaded into the database, its location names are normalized in ASCII format utilizing Unidecode [3] to allow comparison to entries in our registry database. Conversion for Latin-based languages is generally simple and error free, but transliteration of non-latin languages is complex and often ambiguous. Additionally, some location names are incredibly common, such as San Isidro, which is used to name more than 300 locations in the Philippines. Finally, the centroids listed in the registry database inject some ambiguity since they are a limited representation of any location (cities vary greatly in size and layout).

To address these sources of ambiguity we check spatial proximity check between available shapes and the location's centroid. Second, we perform a Levenshtein string comparison to allow for spelling variations caused by transliteration. Matches passing both checks are merged into a single shape. In cases where no city-level match is available the system returns a matching regional shape that maintains correctness at the cost of accuracy.

Sources shapes have vertex counts ranging from hundreds to hundreds of thousands and are a common input to Alidade's intersection calculations. In order to maximize Alidade's scalability and speed all shapes are simplified prior to being loaded on the Hadoop cluster. In order to accomplish this task, we make use of an α-shape [9] algorithm. α-shapes provide an efficient representation that significantly reduces vertex counts, while minimizing the total area added to shapes. Additionally, all generated α-shapes are closed and form concave hulls, so no portion of a city is cropped during simplification. Figure 7 shows the metropolitan area of Neiheim, Germany. The input shape contains 866 vertices. A simple convex hull requires only 32 vertices, but adds significant area to the shape. In contrast, the α-shape depicted consists of only 49 vertices and retains much of the original shape's characteristics. Furthermore, the α parameter can be tuned to achieve any desired tradeoff between complexity and accuracy.

3.9 Additional Shape Sets

We have compiled a large set of shape files for the world's significant bodies of water. As with city and country shapes, these water files must be simplified to allow efficient processing. However, simply applying the same α-shape simplification to bodies of water might expand their area, clipping adjacent land masses. This is problematic since most population centers, and hence target locations, are often concentrated in coastal areas. Thus, any simplification to a water shape must only reduce its area. This can be accomplished by allowing holes in α-shape calculation and then only retaining the interior ring of output shapes for use.

Initial testing of water shape use has revealed that they have the greatest utility for targets with no country or city level hints, since most country and city shapes already incorporate water boundaries. Additionally, we have discovered that a small, but significant portion of tested location predictions from commercial competitors are being placed in coastal waters and that on rare ocassion predictions are much futher from land. Alidade's use of alpha shapes for water areas helps strike a balance between efficiency and accuracy in a manner unavailable to point based geolocation systems.

4. EVALUATION

We evaluated Alidade by comparing its answers with that of six commercial geolocation databases–*EdgeScape*, *MaxMind GeoCity*, *MaxMind GeoCity2 Lite*, *DB-IP*, *IP2Location* and *IPligence*. In this section, we begin with an exposition of the sources of ground-truth location information and the experimental setup. We follow up with a discussion of the evaluation results and show some of Alidade's unique strengths.

4.1 Ground-truth Data

We use six different ground-truth data sets to compare and contrast Alidade's geolocation accuracy with the other geolocation databases. Table 1 summarizes the number of IP addresses available in each data set and the number of unique locations, at approximately 1 km resolution, over which the addresses are distributed. Locations of PlanetLab nodes, referred to as *PLAB*, is a commonly used ground-truth data set in geolocation research. Although data set exhibits good geographic diversity, we show, in Section 4.3 that PLAB does not help to adequately evaluate a geolocation system.

Data Set	#IPs	#Locations
PLAB	835	331
Ark	66	61
MLAB	882	36
GPS	152	139
NTP	99	77
EuroGT	23737281	73

Table 1: Summary of evaluation data sets

The Measurement Lab (*MLAB*) and CAIDA's Archipelago (*Ark*) infrastructure provide a rich ecosystem for networking research, and they both offer a global network measurement platform. The MLAB servers and Ark monitors are located in various countries across the globe offering an interesting diversity in ground-truth locations. In fact, although the Ark monitors are fewer in number they are richer in diversity compared to all other data sets: the 66 Ark monitors are located in 36 different countries, making them an interesting candidate for use in testing geolocation systems.

In Table 1, the term *GPS* refers to a set of IP addreses used by GPS receivers to communicate their locations (and/or measurements) to a base station over the Internet. These GPS receivers are part of a measurement platform employed by geologists to study continental drift. We refer to Network Time Protocol servers with ground-truth location data as *NTP* in the table. The GPS and NTP data sets are ideal for use in testing passive geolocation systems because these data sets enforce a strict policy against active probing.

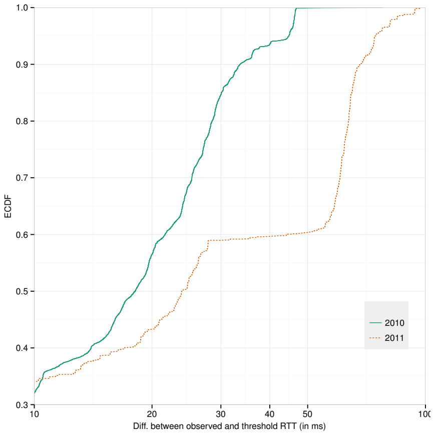

Figure 8: Measurement Errors

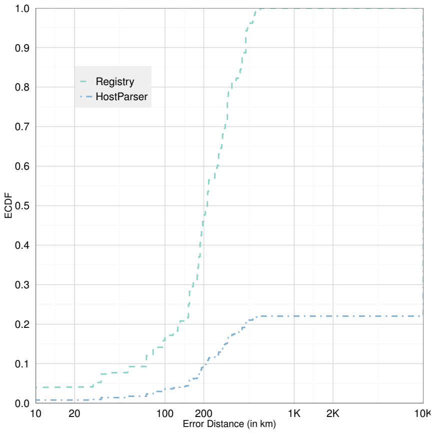

Figure 9: EuroGT Baseline

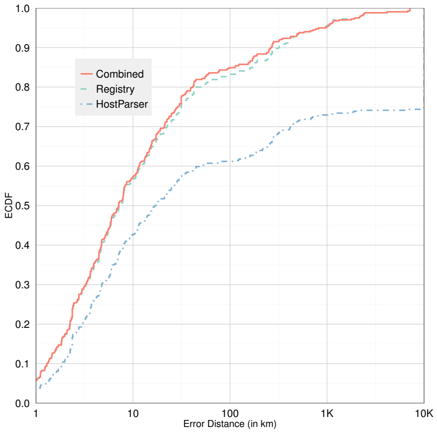

Figure 10: PLAB Baseline

The *EuroGT* ground-truth data is a list of city locations for approximately 24 million IP addresses provided by a European Tier-1 network provider. One pecularity of this data set is that it contains only 73 distinct city locations, although presumably this provider has infrastructure in more than 73 cities. In spite of the relatively low geographic diversity exhibited by this data set, we demonstrate that the data set is still a viable candidate for the head-to-head comparisons of geolocation databases.

4.2 Experimental Setup

The database of IP-address-to-location mappings generated by Alidade was generated from a set of input data sets that included both measurement and non-measurement data. The non-measurement data consisted of HostParser hints for approximately 700 million addresses, of which roughly 211 million contain city-level predictions, location hints compiled from various Internet registries, AS hierarchy data from CAIDA, ground-truth locations of landmarks, and shape files for cities and countries along with accompanying metadata.

Much of the measurement data for the experiment was provided by a Content Delivery Network (CDN) and consisted of traceroutes between CDN servers and hundreds of thousands of resolving DNS servers collected over a period of three months (recorded by the CDN for network mapping purposes), traceroutes from CDN servers to a small fraction of end user addresses collected over a period of three to six months, one week of ping measurements from CDN servers to routers (recorded by the CDN to estimate network performance), and one month of round-trip latency values recorded between CDN servers and end-user machines for a small fraction of TCP connections. The database of results created using these measurement and non-measurement inputs was used as input to the querying engine to geolocate the targets in the evaluation data set. The database contains predictions for approximately 900 million targets generated using these inputs. The selection of targets for evaluation was performed after Alidade's database was finalized; Alidade's results had no influence on selection of targets for the performance comparison.

We used the latest versions, updated in September 2013, of all the databases except for MaxMind GeoCity, for which the last update available to us was made in early June 2013. This is one of the reasons that we have included two databases from the same provider in our study. MaxMind GeoLite2 City, the free version of MaxMind, has also been widely used in academic research for evaluation of geolocation systems. Alidade's input data provided by the CDN is associated with the third and fourth quarters of the year 2013. Our objective is to align simply the different geolocation database systems as closer in timeline as practically possible to ensure a fair evaluation.

We define *error distance* as the geographic distance between a system's point-based prediction for a target and the target's ground-truth location. Although, Alidade outputs polygonal regions as answers, it also computes a point-based estimate, which is *always* contained in the polygonal region. This enables a head-to-head performance comparison of Alidade with the other geolocation databases, all of which provide point-based predictions. Alidade uses various heuristics to output a point-based answer. Picking the center of a city enclosed by the polygonal answer, is an example of such a heuristic.

4.3 Comparative Evaluation Results

We begin by analyzing the effectiveness of relying solely on hints derived from the registry or from the names of the

Data Set	#targets	Coverage
PLAB	289	34.61%
Ark	**0**	**0%**
MLAB	**0**	**0%**
GPS	12	7.89%
NTP	7	7.07%
EuroGT	61,947	61.95%

Table 2: #targets with measurements

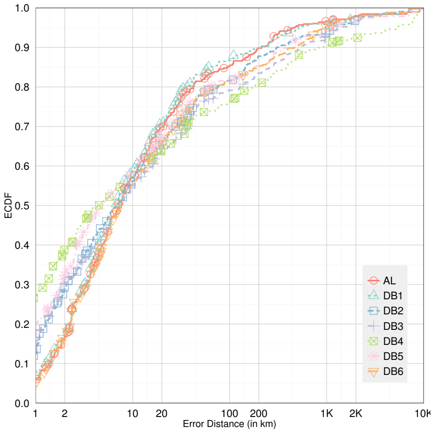

Figure 11: PLAB Results

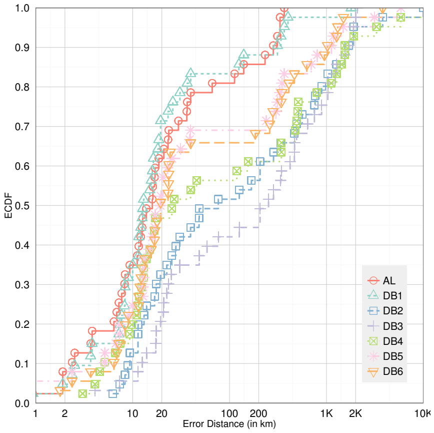

Figure 12: MLAB Results

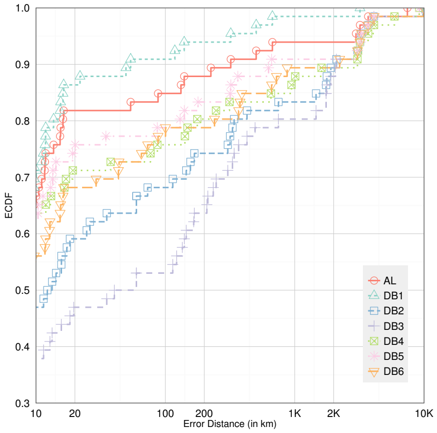

Figure 13: Ark Results

target addresses. These are the primary sources of non-measurement data used by Alidade. Figure 9 shows the ECDFs[1] of errors for the complete 24-million-address EuroGT dataset using only HostParser or registry. HostParser provides answers to just a little over 20% of the targets; for targets with no answers (approximately, 18 million) we assumed an error distance of 10,000km. Registry, by comparison, performs better, with a median error distance of 214km. The results indicate that these two data sources alone are not sufficient to make accurate predictions; in spite of the relatively low geographic diversity in locations EuroGT is still a challenging data set for geolocation.

Many academic studies on geolocation have used PlanetLab nodes as the targets for evaluation, because their ground-truth locations are known (with a few pernicious exceptions). Figure 10 shows that the locations of many PlanetLab nodes can be predicted to a high degree of accuracy using information only from the registry or from HostParser. Marginally better accuracy can be obtained by combining these two data sets. Comparing Figures 9 and 10, we see that the registry and HostParser are much more effective at predicting PlanetLab locations than at predicting targets from our Tier-1 ISP. Hence using these PlanetLab nodes as targets for evaluating geolocation systems that exploit registry information or host names may lead to optimistic results.

For each evaluation we compute the error (distance) in the prediction made by the different geolocation systems for each target in a given data set. We evaluate the different systems by comparing the ECDFs of the computed error distances of each system against the others. To remain consistent with the past geolocation studies, we first evaluate Alidade's geolocation accuracy against the PLAB data set. Figure 11 shows that Alidade's median error distance is slightly higher compared to DB4 and DB5; but, these databases have significantly lower accuracy in the tail portion of the ECDF. For approximately 40% of the targets, Alidade's performs better geolocation compared to the other geolocation databases. DB1 is an exception with its geolocation accuracy being slighly better than Alidade. In this and other data sets, DB1 remains highly competitive with Alidade and exhibits similar performance. We highlight certain key differences, if any and reserve a detailed explanation of DB1's performance to a later section.

Surprisingly, in the PLAB results Alidade's accuracy is only marginally better than the baseline, shown in Figure 10. We presume that this indicates lack of really short measurements to improve upon the baseline estimates. Another plausible reason could be that Alidade's input provides measurements to only 34.61% of the targets in PLAB. Table 2 shows the number and percentage of IP addresses for which some (latency-based) measurement is available in Alidade's input data.

MLAB results, in Figure 12, show that Alidade's geolocation accuracy is significantly higher compared to the other geolocation systems; the median error distance for Alidade is 16km. Alidade's accuracy is relatively lower than that of DB1 in the range from 20-200km. However, Alidade's overall performance is better than DB1 with all targets geolocated within an error distance of 370km – a factor of six smaller than the maximum error distance of DB1. This is in spite of having no measurements whatsoever to any target in the MLAB data set. HostParser and registry provides hints for 5% and 27% of the targets, while the remaining 68% of the targets are geolocated based on the aggregates generated by the aggregator.

The Ark results show, once again, Alidade and DB1 being similar in performance while the rest are approximately one order of magnitude away – 80% of the targets have an error distance of less than 14km when geolocated using Alidade or DB1 compared to an error distance of over 100km with the other systems. The maximum error distance in Alidade and DB1 is around 3200km which is at least three times smaller than the maximum error distances in the other geolocation databases. Recall that Alidade has no measurements, and hence no latency-based constraints to any targets in the Ark

[1] Plots use log-scale for the x-axis, unless mentioned otherwise.

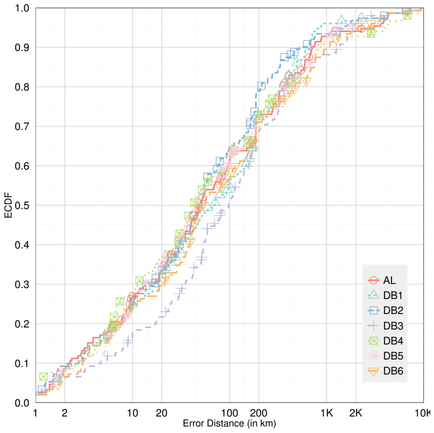

Figure 14: GPS Receivers

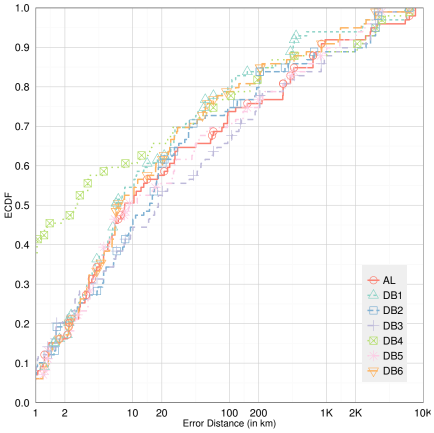

Figure 15: NTP Servers

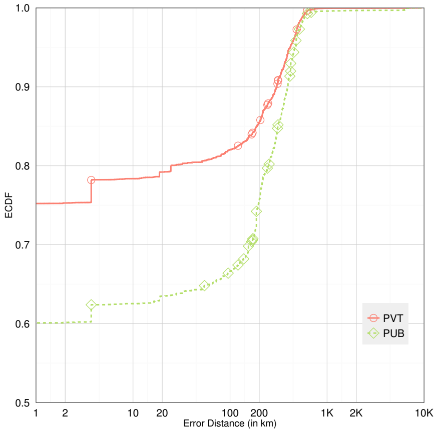

Figure 16: Pub. vs Pvt. data

data set.

For a comparative analysis using the EuroGT data set, we selected a set of 100,000 targets uniformly at random from the data set and evaluated the performance of the different systems against this sample. Figure 1 presents the ECDFs of error distance for each database. Since the ground truth for the EuroGT dataset is only at the city level, we begin the ECDF plots at an error distance of 10km^2. Alidade outperforms all the geolocation databases with 80% of targets located with an error of 10km or less.

Alidade remains competitive in the GPS data set, but has considerably lower accuracy compared to at least three other geolocation databases in the NTP data set. A small fraction – 7-8% – of the targets in this data set contain measurements in Alidade's inputs while the majority have no latency-based measurements. Since we do not know how the different commercial geolocation databases make their location predictions for different targets, we cannot answer how they perform better in this or any other data set. However, we found evidence of "hard coded" answers in one of the widely used commercial geolocation systems. Comparing Alidade answers shows that while Alidade manages to provide better estimates compared to these hard-coded answers, it also loses in some cases by a huge margin. We presume that such hard coded answers might be used in general by all commercial geolocation systems, but do not have evidence to prove it. Alidade, however, has no such hard coded answers.

4.4 Lessons Learned

To gauge the importance of measurement data, we compare ECDFs for those targets for which any kind of measurement data is available (e.g., the target appeared on a traceroute path) with those for which no measurement data is avilable, as shown in Figure 18. We combined targets from all our ground-truth data sets for this analyses. In Figure 18, the curve for the targets with measurements is labeled *WITH-MEAS*, while that for targets without measurements is labeled *WITHOUT-MEAS*. Of the 102,034 targets, measurement data was available for 62,260 targets, while no measurement data was available for the remaining 39,774 targets. The plot confirms the hypothesis that it helps to have measurements in addition to data from the registries or HostParser. Although the improvement from addition of measurements seems minimal, recall that targets without measurements are geolocated using aggregates which might include other targets with measurements. In other words, measurements indirectly influence the accuracy of targets that themselves have no measurements.

In the absence of measurements, aggregates may be helpful for geolocating a target. It is not obvious, however, whether any improvements might result from using aggregates when measurements and other hints are already available for a target. Figure 18 also plots the impact of aggregates on improving geolocation accuracy for targets with measurements and other hints, if any. The *SKIPPED-AGG*-ECDF contains the same set of targets represented by that WITH-MEAS-ECDF, but this time geolocated skipping the use of aggregates. The gap between these two ECDFs highlights the gains from using aggregates on targets already having measurements.

Stale input data can easily cripple a geolocation system. For instance, as the network path between a landmark and a target in the Internet changes, prediction logic like that based on the extrapolator, in Section 3.3, will also change. To demonstate the importance of aligning the input data and ground-truth data sets closer in the timeline, we geolocated the 100,000 targets sampled from the EuroGT dataset using two different sets of input – one gathered from late 2013, and closer in time to when the ground-truth was obtained, and the other from early 2014, one quarter or more away from the ground-truth data collection. Figure 17 shows that the results using input data from 2014 have approximately 20% fewer targets with an error distance of 10km or less.

While measurements may not help in pinning down where

[2] We treat all predictions made with an error distance of less than 10km equally and do not differentiate between them.

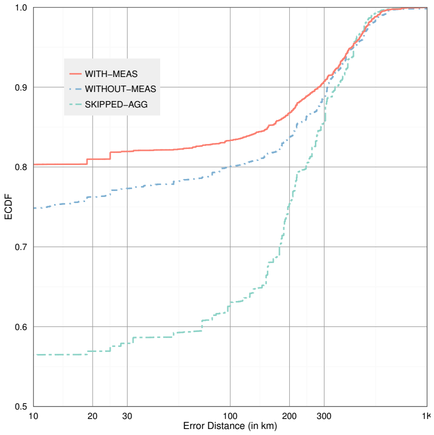

Figure 18: Impact of measurements & aggregates on accuracy

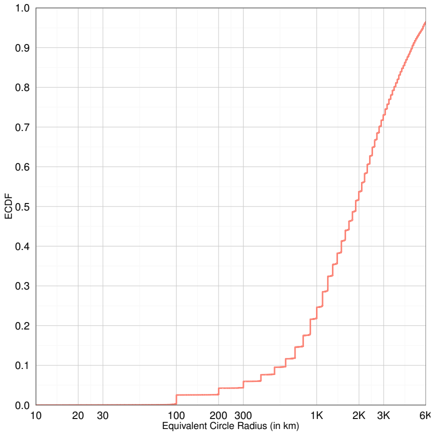

Figure 19: Equivalent circle radii of answer areas

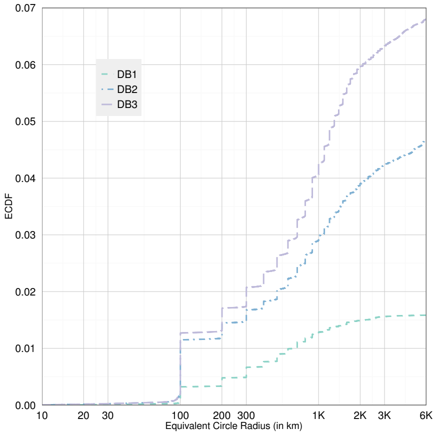

Figure 20: Answers violating direct measurements

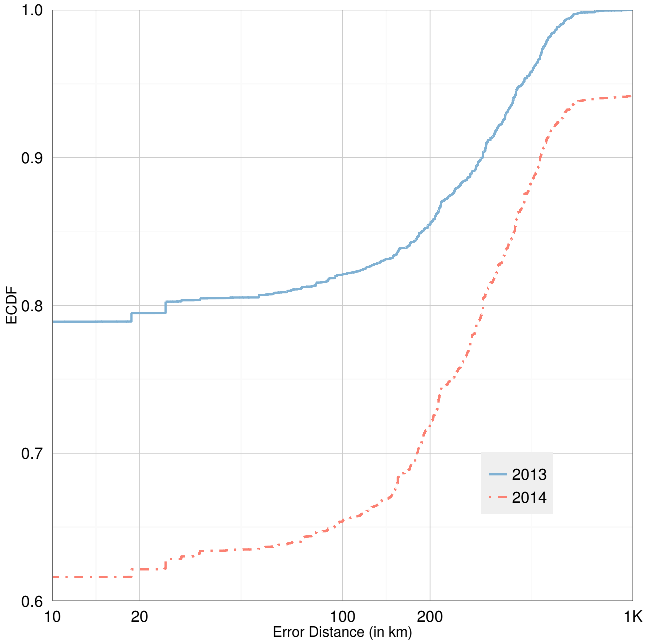

Figure 17: Timeline Alignment Issues

an IP is (except when it is very small), it helps to weed out impossible answers. As an example, Apple has 17.0.0.0/8 and it is likely that most geolocation databases would have the majority (if not all) of the Apple IP space to be in Cupertino, California based on registry information. However, with a measurement data like the one below, it is clear that this particular IP from Apple is somewhere in Asia (and may be in Hong Kong) and definitely not in Cupertino, California. With measurement data, Alidade will produce a feasible area where an IP can be. Such unique geolocation feature from Alidade provides a way to check if the answers from a geolocation DB is incorrect (like the example IP from Apple above). From a recent run, Alidade has feasible answer areas for about 500 million with equivalent radius ranging from a few km to thousands of km, without observable concentration in a particular equivalent answer radius. Using this data, we perform a check against the answers from 3 different commercial geolocation DB and identify the ones that the answer would be outside the Alidade feasible answer area. Figure 19 shows the distribution of the answer area from this run, and Figure 20 shows the results of incorrect answers from various geolocation DB. It can be observed that we can see a non-neglible percentage of incorrect answers from all these geolocation DB and a higher percentage of incorrect answers from geolocation DB for Ips with a relatively small feasible area. Figure 16 shows Alidade's results for targets from all the ground-truth data sets from the inputs from a CDN (PVT) and inputs from iPlane and CAIDA (PUB). We argue that relatively low diversity of landmarks and lack of really short measurements show poor results when using the public datasets.

5. FUTURE WORK

Perhaps the most pressing work that we would like to tackle in the future is to evaluate Alidade against other large ground-truth data sets. The challenge, of course, lies in obtaining such data sets. In addition, several enhancements to Alidade are now in the works. Alidade is already IPv6 compatible, but at present, we do not have much input data related to IPv6. Geolocating mobile devices is a big challenge. Long term, we would like to be able to identify which addresses are being used by mobile devices and, if possible, to estimate the range of locations at which each address is used.

6. CONCLUSION

This paper presents Alidade, a geolocation system that borrows and builds on the best techniques from many previous systems. Unlike nearly all geolocation systems reported in the academic literature, however, Alidade does not perform any active probing on its own, but instead precomputes predictions for all IP addresses prior to fielding any queries. Alidade competes more directly with commercial geolocation databases, and our analysis shows that Alidade is competitive a number of them on six different sets of targets for which ground-truth physical locations are known. While we do not know the details of how the competing databases are compiled, we hypothesize that Alidade makes much more extensive use of network measurement data. Our analysis also shows that while no one source of data suffices to provide very accurate predictions, when these data sources are combined in the right way the overall accuracy can be quite high. We also show that measurement data can be used to filter out geolocation hints from other sources that are not consistent with constraints derived from converting these measurements into distances by multiplying them with two-thirds the speed of light.

7. REFERENCES

[1] Akamai Technologies, Inc. EdgePlatform. `http://www4.akamai.com/html/technology/products/edgescape.html`, 2013.

[2] M. J. Arif, S. Karunasekera, and S. Kulkarni. GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements. In *Proceedings of the Thirty-Third Australasian Conferenc on Computer Science - Volume 102*, ACSC '10, pages 89–98, Darlinghurst, Australia, Australia, January 2010. Australian Computer Society, Inc.

[3] Sean M. Burke and Tomaz Solc. Unidecode, 2013.

[4] CAIDA. Archipelago measurement infrastructure. `http://www.caida.org/projects/ark/`, 2013.

[5] CAIDA. AS Rank: AS Ranking. `http://as-rank.caida.org/`, 02/19/2013 2013.

[6] Nicole Caruso. A Distributed System For Large-Scale Geolocalization Of Internet Hosts. diploma thesis, Cornell University, Ithaca, NY, 2011.

[7] United States of America Census Bureau. TIGER/Line Shapefiles. `http://www.census.gov/geo/www/tiger/shp.html`, 2012.

[8] Frank Dabek, Rox Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *ACM SIGCOMM*, pages 15–26, August 2004.

[9] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the Shape of a Set of Points in the Plane. *IEEE Trans. Inf. Theor.*, 29(4):551–559, September 2006.

[10] Brian Eriksson, Paul Barford, Bruce Maggs, and Robert Nowak. Posit: a lightweight approach for IP geolocation. *SIGMETRICS Perform. Eval. Rev.*, 40(2):2–11, October 2012.

[11] Brian Eriksson, Paul Barford, Joel Sommers, and Robert Nowak. A Learning-Based Approach for IP Geolocation. In *Proc. of the 11th International Conf. on Passive and Active Measurement*, PAM'10, pages 171–180, Berlin, Heidelberg, April 2010. Springer-Verlag.

[12] Michael J. Freedman, Mythili Vutukuru, Nick Feamster, and Hari Balakrishnan. Geographic Locality of IP Prefixes. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 13–13, Berkeley, CA, USA, 2005. USENIX Association.

[13] Phillipa Gill, Yashar Ganjali, Bernard Wong, and David Lie. Dude, where's that IP? Circumventing measurement-based IP geolocation. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 16–16, Berkeley, CA, USA, 2010. USENIX Association.

[14] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-Based Geolocation of Internet Hosts. In *ACM Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004.

[15] Chuanxiong Guo, Yunxin Liu, Wenchao Shen, H.J. Wang, Qing Yu, and Yongguang Zhang. Mining the Web and the Internet for Accurate IP Address Geolocations. In *INFOCOM 2009, IEEE*, pages 2841–2845, 2009.

[16] Robert Hijmans. Global Administrative Areas. `http://www.gadm.org/`, 2012.

[17] IP2Location.com. IP2Location. `http://www.ip2location.com/`, 2013.

[18] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 71–84, New York, NY, USA, October 2006. ACM.

[19] S. Laki, P. Mátray, P. Hága, T. Sebok, I. Csabai, and G. Vattay. Spotter: A Model Based Active Geolocation Service. In *IEEE INFOCOM*, April 2011.

[20] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and kc claffy. As relationships, customer cones, and validation. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 243–256, New York, NY, USA, 2013. ACM.

[21] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An Information Plane for Distributed Services. In *OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation*, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.

[22] MaxMind, Inc. GeoIP City. `http://www.maxmind.com/en/geolocation_landing`, 2013.

[23] Donald R. Morrison. PATRICIA: Practical Algorithm To Retrieve Information Coded in Alphanumeric. *J. ACM*, 15(4):514–534, oct 1968.

[24] Net Industries, LLC. HostIP.info. `http://www.hostip.info`, 2013.

[25] T. S. Eugene Ng and Hui Zhang. Towards Global Network Positioning. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 25–29, New York, NY, USA, November 2001. ACM.

[26] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proceedings of ACM SIGCOMM conference*, San Diego, CA, USA, August 2001.

[27] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2):53–56, April 2011.

[28] Neil Spring, Ratul Mahajan, David Wetherall, and Thomas Anderson. Measuring ISP Topologies With Rocketfuel. *IEEE/ACM Trans. Netw.*, 12:2–16, February 2004.

[29] Yong Wang, Daniel Burgener, Marcel Flores, Aleksandar Kuzmanovic, and Cheng Huang. Towards Street-Level Client-Independent IP Geolocation. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 27–27, Berkeley, CA, USA, April 2011. USENIX Association.

[30] Bernard Wong, Aleksandrs Slivkins, and Emin Gün Sirer. Meridian: A Lightweight Network Location Service without Virtual Coordinates. In *ACM SIGCOMM*, volume 35, pages 85–96, August 2005.

[31] Bernard Wong, Ivan Stoyanov, and Emin Gün Sirer. Octant: A Comprehensive Framework for the Geolocalization of Internet Hosts. In *NSDI*, April 2007.

[32] Inja Youn, Brian L. Mark, and Dana Richards. Statistical Geolocation of Internet Hosts. In *ICCCN*, pages 1–6, 2009.