Towards Geolocation of Millions of IP Addresses*

Zi Hu John Heidemann Yuri Pradkin
USC/Information Sciences Institute {zihu, johnh, yuri}@isi.edu

ABSTRACT

Previous measurement-based IP geolocation algorithms have focused on accuracy, studying a few targets with increasingly sophisticated algorithms taking measurements from tens of vantage points (VPs). In this paper, we study how to scale up existing measurement-based geolocation algorithms like Shortest Ping and CBG to cover the whole Internet. We show that with many vantage points, VP proximity to the target is the most important factor affecting accuracy. This observation suggests our new algorithm that selects the *best few* VPs for each target from many candidates. This approach addresses the main bottleneck to geolocation scalability: minimizing traffic into each target (and also out of each VP) while maintaining accuracy. Using this approach we have currently geolocated about 35% of the allocated, unicast, IPv4 address-space (about 85% of the addresses in the Internet that can be directly geolocated). We visualize our geolocation results on a web-based address-space browser.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network topology*; C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Internet*; C.2.6 [**Computer-Communication Networks**]: Internetworking

General Terms: Experimentation, Measurement

Keywords: IP geolocation, IPv4

1. INTRODUCTION

IP geolocation is the process of finding geographic locations of Internet Protocol addresses. IP geolocation is widely used today. For example, companies use IP geolocation to limit the content to certain countries (for example, television and movies that are often licensed differently by the viewer's country) and to customize advertising based on location. Internet researchers use IP geolocation to relate network phenomena to countries, such as studying the cultural impacts on social networking, or rates of computer crime by country and policy. Moreover, IP geolocation is essential in law enforcement to identify the appropriate jurisdiction to handle enforcement of computer crime statues.

Several research and commercial geolocation systems exist, exploring many different approaches. They form three rough categories (Section 3): systems driven by databases ([11, 15]), measurement based geolocation (such as Geoping [15], CBG [4], and others), and target-assisted geolocation (such as Skyhook [17]). We focus on measurement-based systems here, since they provide better coverage and accuracy than database approaches and are independent of the target. Measurement-based algorithms all depend on *vantage points* (VPs) to actively probe the geolocation *targets.* We study both Geoping- and CBG-like algorithms.

Our goal is not to invent a new geolocation algorithm, but to understand how existing algorithms can *scale up* to many millions of targets and the entire IPv4 address space. We encounter several problems in scaling-up existing algorithms to the whole Internet. First, all existing work uses a relatively small set of VPs, typically tens of VPs [4, 9]. Second, existing work is tested on a relatively small set of targets, typically hundreds of targets. Typical targets are selected with known ground truth to evaluate algorithm accuracy. With dozens of VPs and hundreds of targets, current algorithms each have all VPs send many probes to each target. While this product is reasonable when both are small, with hundreds of VPs and a billion targets, the product is large. The result is a huge amount of traffic out of each VP, burdensome traffic into each target, where hundreds of probes arrive to *each* IP address in a target block, and a heavy load to bring this data together.

To scale geolocation to the entire Internet, our first contribution is *to study what factors affect geolocation scalability and accuracy* for the measurement-based geolocation protocols. We show that *traffic*, both outbound from VPs and inbound to targets, is a significant limitation to full-Internet geolocation, and show that fewer VPs can make inbound traffic manageable. We then show that most VPs provide little benefit to geolocation, suggesting that one can select only a few VPs to geolocate each IP address, getting reasonable accuracy while greatly reducing traffic. We develop three conjectures on factors that affect accuracy and show that good accuracy with a few VPs is possible (Section 4.1).

Our second contribution is to define *new algorithms to choose the right few VPs* (Section 4.2). Our idea is to select the closest VPs to the target, since closer VPs provide

*This work partially supported by the United States Department of Homeland Security contract number D08PC75599 All conclusions of this work are those of the authors and do not necessarily reflect the views of DHS.

IMC '12, November 14–16, 2012, Boston, Massachusetts, USA.

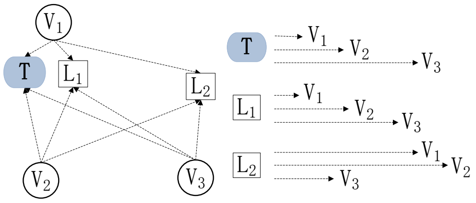


Figure 1: Vantage points (VP), targets and landmarks in Geoping (left), with RTT vectors for Geoping (right).

stronger constraints on location. We show that VP selection using trial measurements to each /24 address block works well (Section 4.2). Our experimental results show that representatives can identify close VPs and provide accuracy almost as good as many VPs. For Shortest Ping, the median error is the same with 10 close VPs compared to all 400 VPs, and for CBG median error is only 11% worse.

With the approaches outlined in this paper we are in the process of geolocating all responsive, public IPv4 addresses. This data is available at no cost to researchers [7].

2. PROBLEM STATEMENT

Our goal is to geolocate every allocated, unicast IPv4 address, with similar accuracy to basic Shortest Ping and CBG. While geolocation algorithms are well known, the main constraint in scaling them up to cover the entire Internet is probing traffic. We next review geolocation elements and these constraints.

2.1 Geolocation terminology

Measurement-based geolocation systems send probes from *vantage points* to geolocate *targets* in the Internet. Some systems also probe *landmarks* at known reference location. Figure 1 shows those entities: V_1, V_2 and V_3 are VPs, T is the target, L_1, L_2 are landmarks.

Targets are public IPv4 addresses that we wish to geolocate. In this paper we assume targets have fixed physical locations, at least over the duration of measurement. Some addresses represent objects that move, such as mobile phones or those using Mobile IP; their identification is future work.

Vantage points (VPs) are hosts in the Internet, always with known locations. They send messages (probes) to targets to determine the targets locations.

Landmarks are IP addresses with known locations used by some geolocation algorithms such as Geoping. Unlike VPs, landmarks do not actively send messages. Since VPs locations are usually known, they often also serve as landmarks.

Address blocks, (or *blocks*), are groups of consecutive addresses, with size determined by the number of leading address bits in common. IP address allocation policies result in blocks that often have common administration [1], often including physical location [15]. We exploit but do not require this assumption.

2.2 Problem Constraints

Unfortunately, it is intractable to probe billions of addresses from hundreds of VPs. Assume we have 500 VPs and each VP probes every IP address 10 times to get the minimum round-trip time (RTT). There are about 3.7 billion allocated, unicast IPv4 addresses. This simple probing from each VP would generate 1.8×10^{13} records (more than 400 TB at 24 B/record), a challenge to process centrally. Other challenges are the amount of traffic to targets and as traffic and processing at the VPs. When combined with the need to probe relatively quickly so that observations can be combined, these challenges motivate our optimizations.

The primary challenge is *incoming traffic to the targets*, both because of its volume and because it can be misinterpreted. While geolocation traffic is not huge, it can be noticed. Ten probes from 500 VPs is only 320 kB of traffic, equal to about a 10 seconds of a Skype video call (at 250 kb/s). While not a huge amount of traffic by itself, the network administrator of a /24 network (or her users) can easily be alarmed at this traffic rate as it covers all 256 addresses over much of an hour. Even a few percent of very vigilant network operators can result in abuse complaints due to concerns of denial-of-service attacks; we must minimize complaints in our use of a shared measurement infrastructure. To avoid complaints, we instead want to pace traffic; we discuss target rates in Section 4.4.

A secondary challenge is sustained, high-rate *traffic at the VPs.* Geolocation requires sustained, symmetric (inbound and out) traffic. At 1 Gb/s, one could cover the geolocatable space in 5 hours, but geolocation requires *geographically* distributed VPs. Current public measurement infrastructure, such as PlanetLab, caps sustained outgoing traffic to less than 10 Mb/s, thus pushing measurement time to more than 500 hours, and longer when the nodes are shared. A simple solution to traffic problems is to spread probes out in time.

Probe pacing must be limited so that measurements are *coherent*, consistent in the face of path or target changes, so arbitrarily slow measurements would be unusable. Probes cannot easily be combined if they cross changes in routing or target movement. According to Paxson, most paths have routes that are stable for days [16]. In our experiment, we assume that most paths are stable for more than two days, so we select a probing rate that allows measurements of each address to complete in at most that time. Detecting and adapting to changes is possible future work.

2.3 Our Approach

To make our goal tractable, we must reduce the workload by using fewer targets or fewer VPs. Prior work such as GeoCluster selects a single target to represent an entire cluster of IP addresses [15], significantly reducing the number of targets. However, our goal is to understand blocks that are geographically homogeneous, and to find blocks that are not.

We therefore instead focus on reducing the number of VPs. TBG shows that geolocation rarely works better than the distance to the nearest landmark [9]. Our approach is based on one similar observation: although one needs *many potential observation points* for accurate IP geolocation, *only a few add information*[1].

In Section 4.1 we formalize this observation as three conjectures and develop a geolocation system around it.

3. RELATED WORK

Three general approaches to geolocation have been proposed. The earliest is *database-driven*, using WHOIS [11, 14], DNS [15], or information from millions of users [10] to infer location, although with generally poor accuracy. Re-

[1] We thank Bill Woodcock of Packet Clearing House for suggesting this observation.

cent work has explored *target-assisted* geolocation, such as with GPS and WiFi-based method such as Skyhoook [17]. These approaches require geolocation code to run on the target, an approach incompatible with covering the entire Internet.

The focus of this paper is on *measurement-based* geolocation: systems that measure network delay from VPs to target to estimate location.

Geoping [15] is based on the assumption that hosts exhibiting similar network delays to other fixed hosts tend to be co-located. In Geoping, all VPs probe many landmarks at known locations, building a set of latency fingerprints (Figure 1). To geolocate an address, all VPs actively probe the target and compare the resulting fingerprint against known landmark fingerprints using Euclidean distance, placing the target at the best matching landmark. Shortest Ping is a simplification of Geoping where targets are mapped to the closest VP (effectively making VPs the only landmarks) [9]. We use Shortest Ping to approximate Geoping because of its efficiency, and its similar accuracy [9].

Constraint-based Geolocation (CBG) instead uses multilateration, where each VP draws a circle with its location as center and the distance (estimated from measured network delay by "bestline" [4]) to the target as radius. CBG locates the target in the overlap of all circles. Topology-based Geolocation (TBG) improves CBG accuracy by considering network topology and using a better latency-to-distance estimate [9]. Octant extends CBG by using both positive constraints (which the target might satisfies) and negative constraints (which the target doesn't satisfy) to reduce the estimated region for the target [18]. We explore basic CBG as representing a second class of algorithms.

Yong et al. propose a three-tiered geolocation algorithm which takes advantage of a massive landmark database and the fact that relative distances are preserved in delay measurements at small scales [20]. In the first tier, the algorithm utilizes the same idea of CBG to geolocate a target IP into a region. In the second tier, the algorithm employs the landmarks in the region of Tier 1 to narrow down the possible region, with the distance constraint-based method. Finally, the target is mapped to the closest landmark found in the region of Tier 2. We do not examine this algorithm because of the difficulty of generating a large landmark database with sufficient detail.

Although we focus on Shortest Ping and CBG as representatives of the two main classes of measurement-based geolocation, we expect that other approaches that also use VPs can also benefit from our evaluation.

4. METHODOLOGY

To address the problem of scaling current geolocation approaches (Section 2), we next describe how selecting a few, good VPs can reduce inbound traffic on each target while preserving accuracy. We first identify three conjectures that must be satisfied for our approach to work, then present specific VP selection algorithms, and finally we describe our overall system to geolocate the world.

4.1 Three Conjectures

To minimize traffic on each target, we probe each target only from a few VPs. In this section we explore three conjectures required to minimize traffic while maintaining accuracy:

1. A few VPs *can be as accurate* as many VPs
2. *Certain small subsets* have good accuracy
3. The *closest* VPs generally maximize accuracy

4.1.1 A Few VPs Can Be Accurate

We begin with our first conjecture: a few VPs *can be as accurate* as many VPs. If all VPs are important to accuracy, then there is no way to reduce traffic.

To evaluate this conjecture we begin with 400 vantage points and randomly select different subsets of 5 to 100 VPs. We do not consider more VPs, because our results are asymptotic with 100 VPs. For each trial we geolocate 25 targets with known locations, using both Shortest Ping and CBG. We repeat each trial 100 times with a different, random subsets of VPs. Our VPs are 400 PlanetLab nodes, while the targets are 25 universities around the world.

Our results depend on the numbers and locations of VPs and targets in several ways. Our premise is that global geolocation should use thousands of VPs and select the best few, so we must study VPs and targets that are both close to each other and distant. To understand the effects of a wide range of topologies, we consider random subsets of VPs (Figures 2c and 3c) and present distributions of accuracy (Figures 2b and 3b). To verify that we do not have degenerate cases (all VPs and targets co-located or distant), we verify that 12 of our 25 targets do not host PlanetLab nodes. Our results represent geolocating the current Internet to the extent that VPs and targets and their topology is representative. Although random samples of VPs present a wide range of topologies, we do not claim to fully represent all Internet topologies. With our targets and many PlanetLab-based VPs at universities, their connectivity is likely better than general Internet connectivity, so our accuracy is likely optimistic for the general Internet (today), with the prediction diverging less as Internet connectivity improves in the future. Ideally, future work would use a more diverse set of targets or VPs, but unfortunately public sources of targets with ground truth are limited today, and growing use of remote hosting makes developing new ground truth a very labor-intensive undertaking.

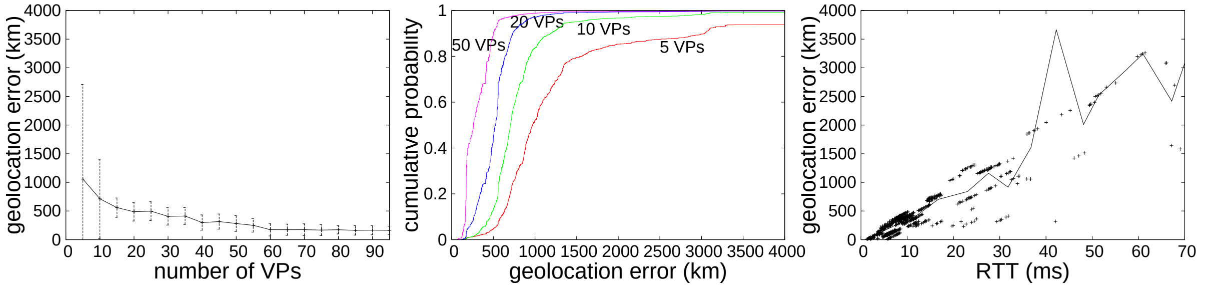
Figure 2a reports median and standard deviations of Shortest Ping accuracy. With more than 60 VPs, median error is small and stable. With fewer than 60 VPs, median error rises and standard deviation is high—some random subsets do well, but some do poorly. This experiment shows that many VPs add no information to geolocation, and that we can reduce the number of VPs, however which VPs are chosen affects accuracy.

Figure 3a shows this experiment repeated with CBG. These results are qualitatively similar to Shortest Ping: moderate numbers of VPs are stable, and use of only a few VPs shows very large variance in accuracy.

4.1.2 Certain Small Subsets Have Good Accuracy

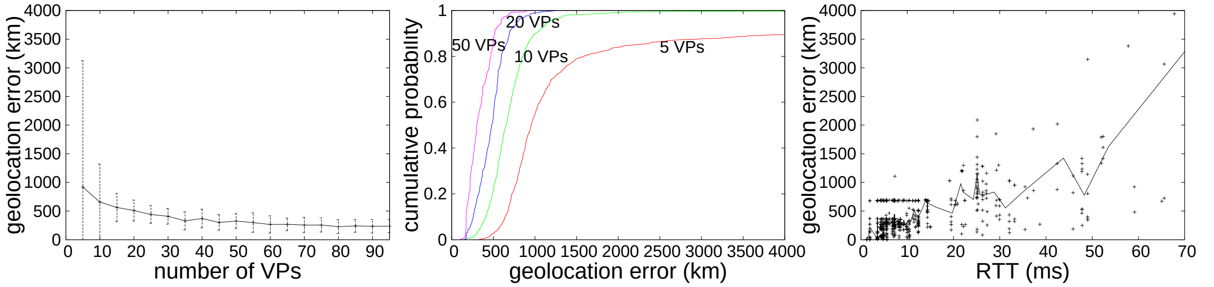
We have shown that subsets can generally have good accuracy, but variance increases greatly as the number of VPs gets very small: some instances do well, but many cases result in large error. We next explore, for a given number of VPs, how different instances perform.

We use the same 400 possible VPs, 25 targets as Section 4.1.1. However, here we look at the distribution of accuracies for several specific sizes of VP subsets. Ideally, for



(a) Number of VPs vs. accuracy (b) Accuracy vs. subset sizes (c) VP distance vs. accuracy (to NYU)

Figure 2: Evaluation of three conjectures for Shortest Ping.



(a) Number of VPs vs. accuracy (b) Accuracy vs. subset sizes (c) VP distance vs. accuracy (to NYU)

Figure 3: Evaluation of three conjectures for CBG.

each size we would study all possible combinations, but that is computationally infeasible. (There are about 83×10^{12} possible combinations of 5 VPs selected from 400.) Instead, we evaluate Shortest Ping and CBG on each of 10,000 randomly chosen subsets.

Figures 2b (Shortest Ping) and 3b (CBG) show the cumulative distribution of median location error across all 25 targets for all 10,000 trials for the two algorithms. The result supports our second conjecture: different instances of same-size VP sets get different accuracy.

With fewer VPs, a small number of cases have very large error (about 5% of cases with 10 to 50 VPs, and 20% of cases with 5 VPs). However, many of the *best instances* have similar accuracy. For example, in Figure 2b, with 5 VPs, the worst instance has errors larger than 4000 km, while the error for the best is less than 42 km, close to the best instance with 50 VPs (error 30 km). (Error with *all* 400 VPs is smaller, at about 12 km, so 80× greater traffic does provide some improvement.) This experiment indicates that if we can select the *right* VP subset, we can achieve fairly good accuracy with just a few VPs.

4.1.3 The Closest VPs Generally Maximize Accuracy

Since we know some combinations of a few VPs provide good results, our final step is to predict which few those are. Our assumption is that *close* VPs provide stricter constraints than far away VPs. To quantify this intuition, we next explicitly compare VP distance and geolocation error. We fix the number of VPs to 10, drawn from the same 400, and geolocate our 25 targets. We carry out 800 trials, each with a different, randomly selected subset of 10 VPs. In each case, we plot the minimum RTT across all VPs with the geolocation error.

Figures 2c and 3c show our study for a single target (a computer at New York University in New York City, USA). This example is representative of all 25 targets. Each point in the figure represents one set of randomly chosen VPs. The line shows the mean geolocation error for all observations in each 5 ms bin of minimum RTTs.

This experiment shows that geolocation error for Shortest Ping has an almost linear relationship with minimum RTT (correlation coefficient 0.88). For CBG, the linear relation between geolocation and minimum RTT holds when the minimum RTT is small (less than 25ms); for larger values of minimum RTT the relationship is much noisier. CBG's correlation coefficient is 0.71 over the entire range. Both graphs show that that small geolocation errors usually have small minimum RTTs, supporting our claim that we can select a good set of VPs by estimating the closest VPs.

4.2 VP Selection and Geolocation

From these conjectures we next propose our VP selection algorithm. We work on /24 blocks, typically consider groups of 6 /8s, processing 150k–217k /24 blocks a time to spread load on the targets (Section 4.4). Although we select VPs based on a few representatives for the block, we geolocate every IP address in the block with those selected VPs to understand how often and which /24 blocks cover multiple locations.

Our algorithm has four steps:

1. Using Internet census histories, select several *representatives* for the block.

2. Probe those representatives from all VPs to *select nearby VPs* for the block.

3. Probe all addresses in the block from nearby VPs to generate *raw geolocation input*.

4. Centralize this input and process it with Shortest Ping (or CBG) to identify *IP geolocation*.

Finding Representatives: We begin by finding representatives for each block. Prior work studied IP hitlists and approaches to select representatives, IP addresses most likely to respond [3]. For geolocation, we *require* at least one representative that responds; we use three to provide some redundancy. If all three representatives do not respond, we just ignore the block. As with hitlist discovery, we use the results of prior IPv4 censuses and the hitlist prediction algorithm to select representatives. Because census histories are public, this step requires no network traffic. We distribute a list of representatives to all VPs for the next step.

Selecting Nearby VPs: To select close VPs, we probe representatives for each block from all VPs. We considered selecting VPs using information about AS paths from BGP to avoid probing from all VPs, but we found it impossible to get BGP data co-located with our 500 VPs. With only a few representatives per block, we probe fairly slowly (200 probes/s) in parallel across all VPs. We probe each representative 10 times, typically taking about 16 hours.

We use our own high-rate probing software, first developed for IPv4 census collection [6]. It probes each address on a target list in a pseudorandom order, spreading probes in time and space to minimize impact on the target networks.

After the representatives are probed, we retrieve all data in parallel to a central site. We estimate VP-representative RTT using the second-to-minimum measurement (discarding the lowest to avoid outliers). Finally, we select the closest VPs for each target block.

Given representatives for each block, we centrally compute target lists unique to each VP and distribute them.

Probing Blocks: VPs then probe targets using the same software. We use a higher probing limit of 500 probes/s, since with 85× (256/3, 3 representatives for each /24) more targets, traffic to each block is still limited. After probing, we copy the raw geolocation data to a central site.

Retrieving Data and Geolocating: Finally, with the raw data centralized, we first extract second-to-minimum RTT for each target, then run standard Shortest Ping and CBG.

4.3 VP Selection and Accuracy

We next examine how our VP selection affects geolocation accuracy. We know geolocation will have some error; our question is: does VP selection increase that error? We randomly select 18 /24 blocks from the ground truth dataset from CAIDA [13], each block with about 100 responsive IP addresses. We then compare the distribution of accuracy of 10 VPs selected by our algorithm against use of all 400 VPs, using both Shortest Ping and CBG.

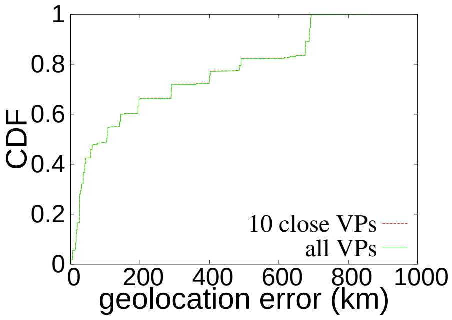


Figure 4: Geolocation error with 10 close vs. all VPs for Shortest Ping

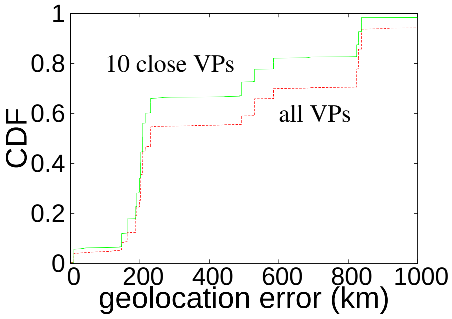


Figure 5: Geolocation error with 10 close vs. all VPs for CBG

Figures 4 and 5 shows accuracy for both Shortest Ping and CBG. For Shortest Ping, the accuracy of our 10 VPs is basically indistinguishable from use of all VPs. Median error is 105 km for both 10 close VPs and all VPs. For CBG, accuracy of using our selected 10 VPS is slightly worse than using all VPs: median error is 231 km instead of 208 km (11% worse). While one would expect this result for Shortest Ping, where accuracy only depends on the closest VP by definition, but we also get similar results for CBG and Geoping. So we conclude that our approach is successful, adding no or only slight error while using only about 2% the number of VPs.

4.4 Managing Probing Rates

Our approach must manage probing rate, going as fast as we can (to maximize coherence) but not so fast as to generate traffic that harms or worries the targets. Our main constraint is the amount of traffic arriving at each target. We evaluate incoming traffic per /24 block, with a goal of being no more than the Internet background radiation [19], that is 0.5 to 0.9 packets/s per /24 block. The traffic rate we generate is:

$$addr\_rate = \frac{probes\_per\_VP \times VPs \times tx\_rate\_per\_VP}{target\_blocks \times targets\_per\_block}$$

$$block\_rate = targets\_per\_block \times addr\_rate$$

Since probes-per-VP and number of VPs are fixed, we control our block-rate by capping the transmit rate and increasing the number of blocks. Our probing software sends probes for a given run in a pseudorandom order, so with a fixed transmit rate, adding targets spreads probes out over more time, decreasing the traffic to each target per unit time.

In our current practice we study 393k /24 blocks per run (6 /8s). For VP selection, we probe only 3 targets per block, so we cap each VPs transmit rate at 200 probes/s. With fewer VPs and more targets per block for geolocation we are able to probe at a greater rate of 500 probes/s, so we estimate that each /24 receives at most 0.64 probes/s.

Experimental verification: The above analysis assumes steady state and smooth flow; we expect traffic will be burstier in practice. To verify our analysis we recorded all incoming probes to one target over 16 hours for VP selection and about 12 hours for geolocation.

Because VPs are not synchronized and probe rates vary due to response, we see that traffic is moderately bursty. If we normalize time to one probe per interval, most periods see no traffic and a few see two or three probes. If we project to block probe rates, the mean observed probe rate is 0.13 probes/s (standard deviation: 0.20) during VP selection, and 0.48 probes/s (standard deviation: 0.79) during geolocation.

Our early use of PlanetLab generated traffic at about double our current rate and drew three complaints from targets. However, after limiting probing as described above, the complaints have stopped.

4.5 Visualization of Geolocation Data

Picturing the results of geolocating the entire IPv4 address space requires new visualization methods. We cannot simply project blocks onto the globe.

Several prior efforts have plotted the IPv4 address space on a Hilbert curve [12, 5, 1, 2]. A Hilbert curve keeps numerically close addresses physically close in two dimensions, and as a fractal it is easy to zoom in or out to control detail. We use it here to show the geographic location of IP addresses, rendering longitude and latitude as color.

As of August 2012, we have geolocated 78 /8s, about 35% of the allocated, unicast, IPv4 address-space. We can only geolocate addresses that respond to probes. From IPv4 censuses we know that those addresses are unevenly distributed; our best estimate is that our progress so far corresponds to about 85% of the addresses in the Internet that can be directly geolocated. Figure 6 shows the map of our geolocation results, and a web-based browser supports pan and zooming down to the individual IP level [8]. Each of the large squares corresponds to a /8 address block (for example, the reddish block near the top left is 2/8). The light green hatched regions are private or multicast address space, the blue hatched regions have not yet been geolocated. Areas that have been geolocated are colored by their location; colors are keyed to latitude and longitude, with hue corresponding to longitude and lightness to latitude, as shown in the country color map at the right bottom of Figure 6. While this color scheme makes some locations similar, we believe a continuous spectrum is important. Code for our color conversion function is freely available at our website.

Because of limited resolution, in this figure each pixel is colored to show the mean latitude and longitude of all addresses in a /18 block. We are evaluating different aggregation methods. While using the mean is our current method, we are considering the use of the modal location, the most common value, although that is sensitive to jitter.

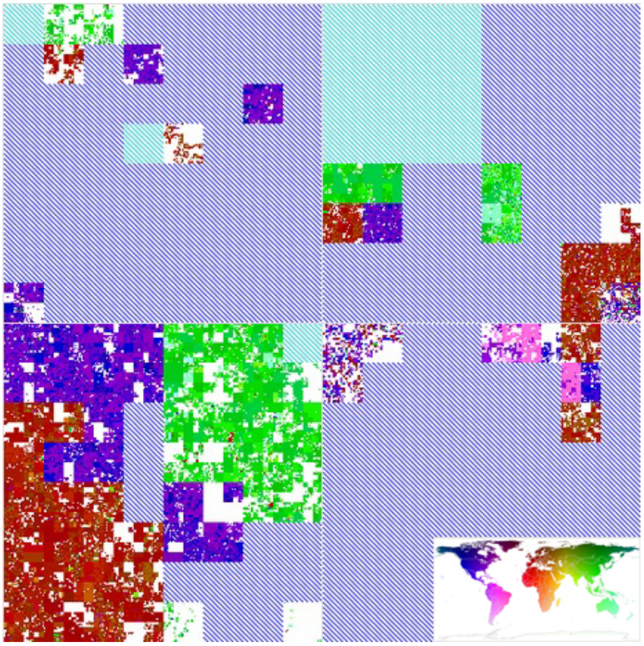


Figure 6: The IPv4 address space placed on a Hilbert curve, with colors corresponding to geolocation (hue follows longitude; lightness, latitude).

5. CONCLUSIONS

This paper studies how to adapt existing measurement-based geolocation algorithms to scale to cover each address in the entire IPv4 address space using many vantage points (VPs). We identify what factors affect geolocation scalability and accuracy, finding that the *right* subset of few VPs can preserve accuracy while greatly reducing traffic. Then we propose a VP selection algorithm to choose these right few VPs for targets, using the closest VPs since they provide the most information. Our experimental results show that the few closest VPs selected by our algorithm are almost as accurate as many VPs, within the limits of our university-centric set of VPs and targets. For Shortest Ping, median error is the same with 10 close VPs compared to all 400 VPs, and for CBG the median error is only 11% worse.

Currently we are geolocating the whole Internet with our proposed techniques. As of August 2012, we have geolocated 78 /8s, about 35% of the allocated, unicast, IPv4 address-space. Visualizations showing geolocation results on a Hilbert curve can be found at our website [8], and our data to-date is available to other researchers [7].

Acknowledgments: We thank Bill Woodcock (PCH) for discussions that prompted this work, Erik Klein (UCLA and USC/ISI) for comments about this work, Kim Claffy and Young Hyun (CAIDA) for discussion of probe rates, the anonymous referees and our shepherd, Vern Paxson, for questions that helped clarify our description. This work was determined to be non-human-subjects research by USC IRB (request IRR00001013, June 2011).

6. REFERENCES

[1] Xue Cai and John Heidemann. Understanding block-level address usage in the visible Internet. In *Proceedings of the ACM SIGCOMM Conference*, pages 99–110, New Delhi, India, August 2010. ACM.

[2] Brian Cort. IPSpace: An interactive visualization of the Internet. unpublished manuscript, October 2006.

[3] Xun Fan and John Heidemann. Selecting representative IP addresses for Internet topology studies. In *Proceedings of the ACM Internet Measurement Conference*, pages 411–423, Melbourne, Australia, November 2010. ACM.

[4] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. Constraint-based geolocation of Internet hosts. *ACM/IEEE Transactions on Networking*, 14(6):1219–1232, December 2006.

[5] John Heidemann and Yuri Pradkin. Mapping the Internet address space. Poster, September 2007. described on the "Mapping the Internet address space" web page `http://www.isi.edu/ant/address/`.

[6] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *Proceedings of the ACM Internet Measurement Conference*, pages 169–182, Vouliagmeni, Greece, October 2008. ACM.

[7] Zi Hu, John Heidemann, and Yuri Pradkin. LANDER geolocation datasets. `http://www.isi.edu/ant/traces/geolocation`, August 2012. Also available through PREDICT (`www.predict.org`).

[8] Zi Hu, Yuri Pradkin, and John Heidemann. Ip geolocation in our browsable IPv4 map. Blog post `http://www.isi.edu/ant/blog/2012/07/02/ip-geolocation-in-our-browsable-ipv4-map/` and Web site `http://www.isi.edu/ant/address/browse/`, July 2012.

[9] 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 ACM Internet Measurement Conference*, pages 71–84, Rio de Janeiro, Brazil, October 2006. ACM.

[10] Maxmind. web page `http://www.maxmind.com/app/ip-locate`.

[11] David Moore, Ram Periakaruppan, and Jim Donohoe. Where in the world is netgeo.caida.org?, July 2000.

[12] Randall Munroe. Map of the Internet. web page `http://xkcd.com/c195.html`, December 2006.

[13] Freebox ADSL networks. IP geolocation ground truth. web page `http://francois04.free.fr/liste_dslam.php`.

[14] University of Illinois. IP to latitude/longitude server. web page `http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll`.

[15] Venkata N. Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *Proceedings of the ACM SIGCOMM Conference*, pages 173–185, San Diego, California, USA, August 2001. ACM.

[16] Vern Paxson. End-to-end routing behavior in the Internet. *ACM/IEEE Transactions on Networking*, 5(5):601–615, October 1997.

[17] Skyhook. web page `http://www.skyhookwireless.com/`.

[18] Bernard Wong, Ivan Stoyanov, and Emin Gün. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation*, pages 313–326, 2007.

[19] Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Houston. Internet background radiation revisited. In *Proceedings of the 10th ACM Internet Measurement Conference*, pages 62–73, Melbourne, Australia, November 2010. ACM.

[20] Wang Yong, Burgener Daniel, Flores Marcel, Kuzmanovic Aleksandar, and Huang Cheng. Towards street-level client-independent IP geolocation. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, Berkeley, CA, USA, 2011. USENIX Association.