Computer Science
Technical Report

Colorado State University

Assessing Co-Locality of IP Blocks

Manaf Gharaibeh
Colorado State University
gharaibe@cs.colostate.edu

Han Zhang
Colorado State University
zhang@cs.colostate.edu

Christos Papadopoulos
Colorado State University
christos@cs.colostate.edu

John Heidemann
University of Southern California/ISI
johnh@isi.edu

November 20, 2015

Colorado State University Technical Report CS-15-103

Computer Science Department
Colorado State University
Fort Collins, CO 80523-1873

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: http://www.cs.colostate.edu

Assessing Co-Locality of IP Blocks

Manaf Gharaibeh[1], Han Zhang[1], Christos Papadopoulos[1], and John Heidemann[2]

[1] Colorado State University, CS Dept., Fort Collins, CO, USA
{gharaibe,zhang,christos}@cs.colostate.edu
[2] University of Southern California/ISI, Los Angeles, CA, USA
johnh@isi.edu

Abstract. Many IP Geolocation services and applications assume that all IP addresses with the same /24 IPv4 prefix (a */24 block*) are in the same location. For blocks that contain addresses in very different locations (such blocks identifying network backbones), this assumption can result in large geolocation error. This paper evaluates this assumption using a large dataset of 1.41M /24 blocks extracted from a delay measurements dataset for the entire responsive IPv4 address space. We use hierarchal clustering to find clusters of IP addresses with similar observed delay measurements within /24 blocks. Blocks with multiple clusters often span different geographic locations. We evaluate this claim against two ground-truth datasets, confirming that 93% of identified multi-cluster blocks are true positives with multiple locations, while only 13% of blocks identified as single-cluster appear to be multi-location in ground truth. Applying the clustering process to the whole dataset suggests that about 17% (247K) of blocks are likely multi-location.

Keywords: Geolocation, Co-location

1 Introduction

Internet location-aware applications and research benefits from IP-to-geolocation provided by services such as MaxMind [5], IP2Location [3], and DB-IP [1]. These services provide various degrees of geolocation accuracy. While the accuracy has improved over the last decade [22], [21], [15], IP-geolocation is still seen as an open problem, with focus on improving precision to the level of cities and street. Good accuracy can be difficult when some blocks of adjacent IP addresses span large geographic areas [19], [11]. Today, most services assume that the addresses in a block with the same /24 prefix (a */24 block*) are usually geographically proximate—the *block co-locality assumption*. When this assumption is violated, some addresses in the block will have poor accuracy.

Free databases such as MaxMind *GeoLiteCity* [5] and IP2Location *LITE-DB11* [3] assume block co-locality. These databases identify blocks of various sides, each with a specific location. MaxMind's database contains about 1.7M block entries covering 3.6 billion addresses (97% of the allocated address space). The IP2Location database has 2.2M entries covering the entire IPv4 address space (although they do not assign locations to special blocks such as multicast). In these databases, nearly all IP addresses show block co-locality—99% of MaxMind and all of IP2Location.

Some location-aware applications also seem to follow the /24 co-locality assumption. An architecture proposed by Chen et al. [8] maps client's request to proximal content server at the level of prefixes. All clients within the same prefix are mapped to the same content server. They suggest the mapping at /20 prefixe granularity to minimize the number of required mappings. The underlying assumption is that clients within such prefixes would experience similar network delays. On the other hand, they study the geographic properties of different prefix lengths using geolocation data from from Akamai's EdgeScape database. They compute the cluster radius of clients in a prefix as the mean of the client's distance to the cluster centroid. Almost all the /24s prefixes have a cluster radius distance of 10 miles or less. These small radius distances in /24 blocks suggest that Akamai's EdgeScape database agree with the block co-location assumption.

In this paper we assess the assumption of co-locality in /24 blocks that exists in many common geolocation databases. We leverage a subset of the dataset collected by Hu at al. [14] and publicly available [7]. The dataset contains round-trip estimates for every responsive address in the IPv4 address space measured from several vantage points (VPs). We assess co-locality in this dataset based on the observation that geographically co-located hosts will show similar network delays when probed by the same set of VPs [17]. Based on this observation, we cluster responsive addresses in each /24 block into groups by similarity of the delay measurements from multiple VPs. We then identify /24 blocks with multiple clusters and show that these clusters violate the block co-locality assumption and likely contain addresses in distinct geographic locations.

Our first contribution is to introduce and evaluate this methodology to assess co-locality of endpoints in a block. In this paper we limit our study to /24 blocks, but the methodology is independent of block size. We evaluate a delay-based clustering algorithm that automatically identifies blocks that appear to have endpoints at different locations. We validate the accuracy of this method via a carefully selected set of /24 blocks that we believe are co-located. We confirm that 93% of multi-cluster blocks identified in the ground truth datasets described in Section 4 are true positives. Our second contribution is the application of the methodology to analyze a dataset of 1.41M /24 blocks (118M addresses). We find that a large fraction of these blocks (17%, or 247K blocks) appear to have endpoints at multiple locations.

2 Dataset

Our analysis uses the ISI geolocation dataset [7] extended from prior work by Hu et al. work [14]. The original dataset contains round-trip time measurements for all the allocated and responsive IP addresses in the IPv4 address space at the time it was taken. The dataset has about 472M IPs in just less than 3.5M /24 blocks and was collected from Feb. 2012 to Mar. 2013. RTTs were measured from about 670 vantage points (VPs), all PlanetLab nodes. The work used an algorithm to pick the 10 closest VPs to any /24 block. Use of VPs close to the target minimizes interference from congestion and maximizes the precision of geolocation (something 400 ms away can be anywhere on earth, but something within a few ms is likely in the same city). To avoid estimating latency due to congestion latency was reported as the minimum of 10 measurements We refer readers to [7] for more details on the original dataset. For our work, we extracted the raw probing data for all /24 blocks with at least 10 IP addresses that responded to all VPs probes. The delay measurements of each IP address are treated as its coordinates in a multidimensional space. Our extracted dataset comprises **118.5M** IP addresses in **1.41M** /24 blocks.

3 Methodology

3.1 Identifying Multi-Location Blocks

Our methodology is based on the insight that geographically co-located IP addresses in a block exhibit relatively similar network delays when probed from the same reference points. We cluster IP addresses in a block based on the similarity of their observed delay measurement from a number of VPs. Each IP address is represented as a vector of its delay measurements. We formulate the problem as finding similar IP addresses in a multidimensional space of delay coordinates. Co-located IP addresses are expected to have small distances in the multidimensional space.

We use an agglomerative hierarchal clustering algorithm from **R** *cluster* package called *agnes* to generate hierarchal structures (dendrograms) for one block IP addresses. We set the clustering method to use standardized Euclidean distance metric to measure IP addresses dissimilarities. A dynamic tree cut method from *dynamicTreeCut* package [16] is used to identify the clusters in the dendrogram. The combination of the these methods suits our need to identify clusters automatically without a prior knowledge of their number or size.

As in other agglomerative hierarchal methods, the *agnes* method generates a hierarchal structure for the input observations bottom-up. Each observation starts as a cluster by itself. In each subsequent step, the closest two clusters not already in the same cluster are merged into one larger cluster. The process continues until there is only one cluster of all observations. The dissimilarity between two clusters can be computed in different ways. In this work we use *average linkage* method. This is the average of pairwise dissimilarities between the objects in the two clusters. For two clusters A with n_a objects, and cluster B with n_b objects, this is computed using Equation 1, where D is the distance metric used to compute the distance between two objects. We use the *Standardized Euclidean* distance metric to balance the depth of the measurements observed from VPs at different distances from targets.

$$d_{average}(A,B) = \frac{1}{n_a n_b} \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} D(IP_{Ai}, IP_{Bj}) \tag{1}$$

Prior work has shown the need for selecting clustering thresholds dynamically when examining Internet RTT data [10]. To identify clusters automatically for each of our 1.41M /24 blocks, we use the "Dynamic Hybrid" tree cut method [16] to dynamically identify clusters in a dendrogram. This method uses dendrogram merging information to build the clusters in a bottom-top fashion. We tuned the method parameters to make conservative restrictions on what to be considered a cluster. For example many of the parameters are set as a fraction of the joining heights of the branches in the dendrogram. The one parameter we found most effective was the *minimum gap* parameter, which specifies the minimum joining height to allow two clusters to be merged. Higher settings of this parameter allow more clusters to be merged. This means fewer clusters with significant differences between them. We favor the higher settings in order to minimize the false positives, and to identify groups that are very different from one another indicating higher chances of being at different locations. We also set the minimum cluster size to 10 to reduces chances of getting small clusters of outliers. We evaluate the clustering method using two ground truth datasets in Section 4.

3.2 Methodology Limitations

Our methodology for clustering a block endpoints has some limitations. First, as all delay-based methods, our approach can be affected by inaccurate measurements. This problem is alleviated by taking multiple measurements over time, also of the use of multiple VPs per block provides more reliability. Another limitation is that our method does not tell how far the identified clusters are from each other. However, this is not the goal of this work. Our main contribution is to provide an automatic method to assess blocks co-locality.

4 Validating Identification of Multi-Location Blocks

We next validate our methodology to show that it accurately finds multi-location blocks in ground truth. We build ground truth, beginning by identifying single-location blocks (Section 4.1), then using this data to construct a multi-location dataset (Section 4.2). We use this ground truth to validate our approach (Section 4.3). Finally, we estimate a false positive upper bound for the clustering method (Section 4.4).

4.1 Building Single-Location Ground Truth Dataset

We first build a dataset of /24 blocks that we believe are single-location blocks. This dataset is used for two purposes: (a) to evaluate the clustering method accuracy on single-location blocks, (b) to build the multi-location ground truth dataset.

Academic institutions typically have a specific, well-defined physical locations necessarily operate many end-user computers in those locations, and often self-host web services [21]. We therefore identify the locations and IP addresses of blocks containing the main websites of 4650 universities from different locations around the world from [6].

We verify this dataset to confirm our assumptions, checking for self-hosting we apply two filters to make sure the corresponding websites are locally hosted at their universities. First, we detect outsourcing using *whois* information and discard outsourced blocks. We identify outsourcing by matching the *OrgName* field with the institution name. For example, Duke University's website is at an IP address (54.191.241.8) which the whois OrgName identifies as *Amazon Technologies Inc*, showing outsourcing. Second, the use Google Maps Geocoding API [2] to identify a university's physical location (latitude/longitude), and compare this to the MaxMind's physical location assigned to the IP address. We then discard IP addresses where the great circle geographic distance between these geographic locations is more than 10 miles. While we recognize these data sources as incomplete, this step discards blocks that are known to differing or uncertain locations. The final step is to look up the remaining IP addresses' /24 blocks in our extracted dataset of 1.41M blocks. We extract the raw probing data for the blocks we find to form our single-location dataset.

After these filters and look ups, we reduce our initial set of 4650 academic institutions to we ended up with only 85 /24 blocks from as many unique institutions with strong confidence in their location.

4.2 Building Multi-Location Ground Truth Dataset

We next build a dataset that contains blocks that are in multiple locations to evaluate our clustering methods. Since there is no public ground truth of block locations (much less multi-location blocks), we create this dataset by combining two blocks at two different locations and treat them as one block.

To generate artificial multi-location blocks we find all blocks from the single-location dataset that are probed by the same set of VPs (*VP-compatible blocks*). We then compute all two-block combinations in each set of VP-compatible blocks, combining all measurement data from the two blocks to create a new artificial block. (Merged blocks may have up to 512 addresses, although since we have data only ping-responsive addresses they almost always have far fewer.) We then treat this combined block as if it was a block of adjacent addresses to evaluate our clustering method.

Some of our single-locations are actually quite close to each other. We therefore identify two subsets of our multi-location dataset: those composed of *almost co-located* locations within 10 miles of each other, and those that are further, all 22 miles distant or more. We identify 21 almost-co-located artificial blocks and 99 proper multi-location blocks.

4.3 Validating Delay-based Identification of Multi-Location Blocks

These ground truth datasets let us validate the corrects of our delay-based clustering method to identify blocks that span multiple geographic locations. We test our method on both single-location and multi-location datasets.

We first consider our single-location dataset. Our clustering algorithm classifies 91% of blocks in our single-location dataset correctly (77 of the 85 /24 blocks are correctly identified as single-location.) Seven blocks are identified to have 2 clusters, and one block was not clustered—none IP addresses met our threshold to identify common locations.

We next turn to our artificial multi-location dataset. Before evaluation, we discard artificial blocks built from the 7 misclassified single-location blocks (since we know those will be identified as multi-location). We then apply clustering to the remaining 99 artificial blocks. Fig. 1 shows the number of identified clusters and the corresponding distance between combined blocks for each combination. We correctly identify 88% of these as multi-location blocks, with 12% false negatives.

To examine the most challenging blocks we also looked at the 21 almost co-located artificial blocks (where the real-world distance of the two parts of each block is within 10 miles). In spite of this close physical, distance we correctly identify 38% of these blocks as multi-location (8 of 21).

Overall, 93% of the cases identified as multi-cluster blocks are true positives in our ground truth datasets (of same and clearly multi-location blocks), giving us confidence that a block is identified to have endpoints at multiple locations.

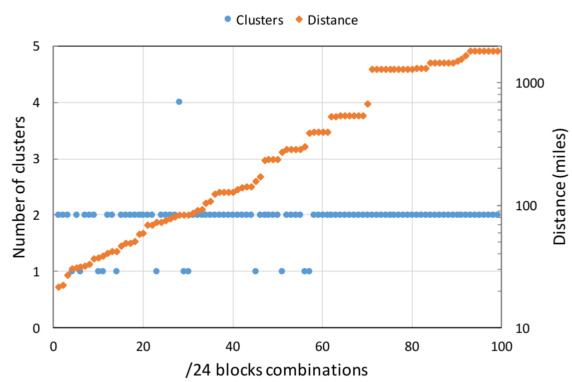

Fig. 1. Results of applying the delay-based clustering method to 99 2-blocks combinations. 88% of the combinations are correctly identified as multi-location blocks.

4.4 Clustering Method False Positives Upper Bound

It is important for our clustering method to maintain a low false-positive rate to insure that we don't overestimate the number of multi-location blocks (false positives). To estimate an upper bound for the false positives rate, we build another extended set of /24 blocks that are likely co-located. The assumption is again, /24 blocks in academic institutions are very likely co-located. Similar to the procedure of collecting the single-location dataset, we collect 100 universities

/16 blocks and verify they do not include web hosting services. Of these 100 /16 blocks there are 3,062 /24 blocks that have latency measurements in our dataset. We run our clustering method on all these blocks. The results show that 239 blocks (7.8%) are not clustered, 2657 blocks (86.77%) have one cluster, and 166 blocks (5.4%) have 2 clusters. Since any of the blocks identified as multi-location could be indeed multi-location blocks, we consider the 5.4% as an upper-bound false positives rate for our clustering method.

5 Co-Locality of /24 Blocks

5.1 Identifying Multi-Location /24 Blocks

In this section we discuss the results of applying our clustering method to the whole dataset of /24 blocks. Fig. 2 shows the distribution for the number of clusters identified for 1.41M /24 blocks. About 17% of them (246647 blocks) appear to have endpoints at multiple locations. 84% of the multi-cluster blocks are grouped into 2 clusters of IP addresses. On the other hand, a very small fraction, 0.7%, of the multi-location blocks are grouped into 5 or more clusters. Our method declined to cluster 73792 (or 5.3%) /24 blocks. 98% of them have 20 IP addresses or less. Basically this means the clustering method could not merge smaller clusters to form a cluster that satisfies clustering criteria. While this is more typical in blocks with small number of IP addresses, it can also be true for any block with endpoints that are highly scattered geographically.

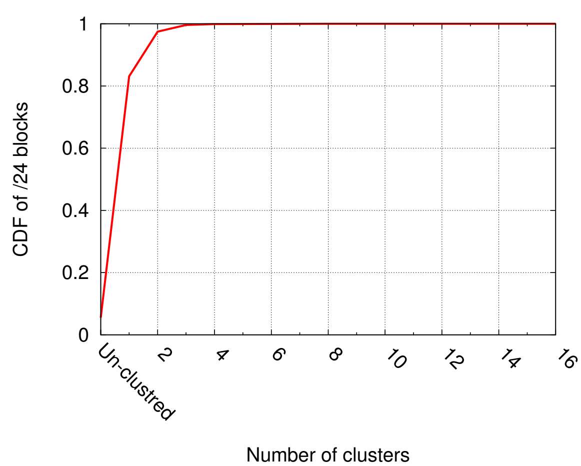

Fig. 2. Distribution of the number of clusters for all 1.41M /24 blocks. More than 17% (247K blocks) are identified as multi-location blocks.

5.2 Comparison with MaxMind *GeoLiteCity* Database

We compare our results with MaxMind *GeoLiteCity* free geolocation database [4]. It is important to note that we are comparing the results on our 3 years old dataset with a recent geolocation database. While some blocks could have been moved or reallocated, we believe the overall picture has not change enough to invalidate the comparison. Table 1 compares our clustering results to the number of locations found MaxMind's *GeoLiteCity* for each /24 block in our dataset. *One to one* case is where we agree that the blocks are single-location, while *many to many* case shows we agree on about 13K blocks as multi-location blocks, which is only about 5% of the 247K blocks we identify as multi-location blocks. The results also show, interestingly, that MaxMind *GeoLiteCity* assigns 2 or more locations to 65K blocks that we identify as single location blocks.

Table 1. Number of clusters to number of locations in MaxMind *GeoLiteCity* database for all blocks in our dataset. 95% of the blocks we identify as multi-location (16.6% of our dataset blocks) are assigned a single location in *GeoLiteCity* database.

Clusters to locations	% of our dataset blocks
One to one	72.7
One to many	4.6
Many to one	16.6
Many to many	0.93
Un-clustered	5.2

Out of the 65K blocks MaxMind assigns multiple locations, 8056 are cases where one of the assigned locations is a coarse country level granularity location. For the remaining cases, we compute the max distance of the block as the max *Great-circle distance* between any two assigned locations' coordinates. Fig. 3 shows the distribution of the computed blocks' max distance for the cases where MixMind assigns multiple locations to a block. The graph show two lines, one is for (one to many) cases, the other is for (many to many cases). Based on *GeoLiteCity* location information the results show that about 86% and 60% of the (one to many) cases involve locations separated by less than 30 miles and 10 miles respectively. While most of the multi-location cases in *GeoLiteCity* involves blocks with small max distance, the clustering method is able to catch 13K cases of them. The red line shows the max distance distribution for these (many to many) cases.

5.3 Characterizing Multi-Location Blocks

Our method identifies about 247K blocks as multi-location blocks. We found multi-location blocks in 182 different countries in our dataset. We list the top 10 countries sorted on the number of multi-location /24 blocks in Table 2. The list is dominated by countries with rich Internet infrastructure like the United States, Japan and western Europe countries. The list comprises about 79% of the total

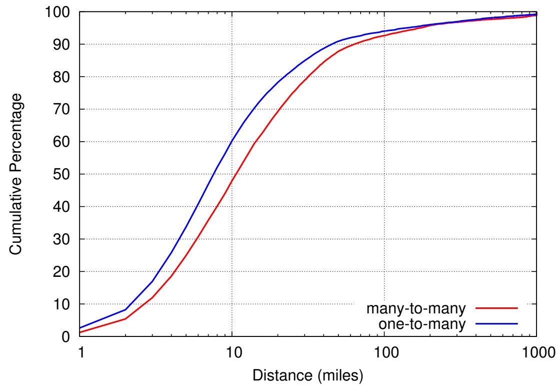

Fig. 3. CDF of max distance for multi-location blocks in MaxMind *GeoLiteCity* computed as the maximum distance between any 2 assigned locations. Interestingly, 86% of the (one to many) cases are for blocks with max distance of less than 30 miles.

number of blocks we identify as multi-location. The list also show the number of unique ISPs within a country with at least one identified multi-location block. Table 3 shows more details for the top 10 ISPs also sorted on the number of identified multi-location blocks. We believe the high numbers of identified multi-location blocks in these ISPs reflect liberal policies of IP addresses assignment to customers with respect to their geographic locations.

Table 2. Top 10 countries sorted on the number of multi-location blocks. The list comprises 79% of blocks identified as multi-location blocks in our dataset.

Country	Num. of blocks	Num. of distinct ISPs
US	83980	2054
DE	34489	285
JP	20834	165
GB	12305	342
KR	10216	103
MX	7761	36
PL	6952	241
FR	6827	164
BR	4748	239
NL	3780	142

6 Related Work

Many of the works in the area of IP geolocation focus on improving geolocation accuracy [13], [15], [22], [21], [9]. The proposed approaches use different techniques but are typically delay-based. These approaches are typically evaluated

Table 3. Top 10 ISPs sorted on the number of multi-location blocks, and their corresponding ASNs and countries. The list is dominated by large ISPs in countries with rich Internet infrastructure.

ISP name	Num. of blocks	ASN	Country
DTAG Deutsche Telekom AG	21204	3320	DE
COMCAST-7922 - Comcast Cable Commu.	11804	7922	US
OCN NTT Communications Corporation	9204	4713	JP
ATT-INTERNET4 - AT&T Services	8994	7018	US
Uninet S.A. de C.V.	7033	8151	MX
UUNET - MCI Communications Services	6881	701	US
CENTURYLINK-US-LEGACY-QWEST	6766	209	US
BSKYB-BROADBAND-AS Sky UK Limited	5810	5607	GB
VODANET Vodafone GmbH	5665	3209	DE
TPNET Orange Polska Spolka Akcyjna	5561	5617	PL

on a small number of targets in the order of few hundreds. Our work does not propose a new algorithm to improve geolocation, and is not limited to a small set of targets. We characterize co-locality of over than 1.4M /24 blocks showing that many appear to have endpoints at different locations.

Other IP geolocation works studied public and commercial databases accuracy and granularity. Poese et al. [18] found that some databases split ISP blocks into smaller ones for more accuracy, however, that made their geolocation accuracy worse. Siwpersad et al. [19] studied the geographic resolution of geolocation databases. They compared location information provided by the databases with locations computed using Constraint-Based Geolocation (CBG) [13]. They concluded that the resolution of the databases is way coarser in comparison. Gueye et al. [12] also used CBG to estimate the max distance between block endpoints to estimate its geographic span, which the concluded could be large. Overall, these works are concerned with geolocation databases accuracy and granularity. The block sizes studied are relative to what is found in the databases. Our work focuses on studying the co-locality of /24 blocks. We use a different methodology that enables automatic identification of groups of similar IP addresses. Compared to these works, our dataset is much larger and representative, and is more recent.

Freedman et al. [11] studied IP prefixes geographic characteristics and its influence on BGP routing table. There results show about 1.4% of /24 blocks or smaller span distances of more than 100 miles. They extract locations of IP addresses based an DNS naming heuristics using *undns* tool [20]. DNS IP to location mapping has many shortages and can be unreliable due to the lack of naming standards, also when the target is unreachable. Our method to study geographic properties of a block is not dependent on IP addresses locations, instead we study their proximity to each other using their latency measurements.

Fan et al. [10] studied the dynamics of mapping users to Front End (FE) clusters, which are groups of geographically close content servers used by Content

Distribution Networks (CDNs). They enumerate CDNs FE servers and then use a clustering technique similar to ours in order to group the servers into FE clusters. While both works use similar delay-based clustering techniques, our studies have different purposes. We use clustering to study blocks co-locality as opposed to their goal of identifying FE cluster of one CDN.

We finally compare our work to Hu et al. [14] work, from which we leveraged our dataset. Hu et al. implemented a method to scale existing delay-based geolocation approaches such as Shortest Ping and CBG to geolocate all responsive IPv4 address space. They show that careful selection of a small number of VPs can maintain a comparable level of accuracy to that when using tens of them. While we use a large subset of their raw probing dataset, the problem we are addressing is different. Their work uses delay measurements to geolocate IP addresses, while we use them as signatures to identify groups of similar endpoints in a block.

7 Conclusions

Our work introduces a simple clustering methodology to assess IP blocks co-locality. We identify groups of IP addresses that appear to be at different location based on their delay measurements observed from a number of VPs. We use a large dataset of 1.41M /24 blocks and show that more than 17% of them appear to be multi-location blocks. This outcome disagrees with the common assumption of IP addresses co-locality in /24 blocks. We also find that the majority of the blocks identified as multi-location belong to large ISPs in rich Internet infrastructure countries like the United State and Western Europe countries. Such large ISPs are apparently not strict about assigning their IP address space to customers within close geographic areas.

Acknowledgments. The work in this paper is partially sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, HSARPA, Cyber Security Division, via SPAWAR Systems Center Pacific under Contract No. N66001-13-C-3001 (all authors), and via BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement numbers FA8750-12-2-0344 and FA8750-15-2-0224 (Papadopoulos and Heidemann). The U.S. Government is authorized to make reprints for Governmental purposes notwithstanding any copyright.The views contained herein are those of the authors and do not necessarily represent those of DHS or the U.S. Government.

References

1. The DB-IP database. `https://db-ip.com` (2015)
2. The Google maps geocoding API. `https://developers.google.com/maps/documentation/geocoding/` (August 2015)
3. IP2Location. `http://www.ip2location.com` (2015)

4. Maxmind GeoLite databases. http://dev.maxmind.com/geoip/legacy/geolite/ (July 2015)
5. Maxmind, inc. https://www.maxmind.com (2015)
6. Universities worldwide. http://univ.cc (July 2015)
7. USC/LANDER project. Internet addresses geolocation dataset, PREDICT. https://www.isi.edu/ant/lander (2015)
8. Chen, F., Sitaraman, R.K., Torres, M.: End-user mapping: Next generation request routing for content delivery. In: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. pp. 167–181. SIGCOMM '15, ACM, New York, NY, USA (2015)
9. Eriksson, B., Barford, P., Maggs, B., Nowak, R.: Posit: a lightweight approach for IP geolocation. SIGMETRICS Perform. Eval. Rev. 40(2), 2–11 (Oct 2012)
10. Fan, X., Katz-Bassett, E., Heidemann, J.S.: Assessing affinity between users and CDN sites. TMA, 95-110(2015) (2015)
11. Freedman, M.J., Vutukuru, M., Feamster, N., Balakrishnan, H.: Geographic locality of IP prefixes. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. pp. 153–158. IMC '05, USENIX Association, Berkeley, CA, USA (2005)
12. Gueye, B., Uhlig, S., Fdida, S.: Investigating the imprecision of IP block-based geolocation. In: Proceedings of the 8th International Conference on Passive and Active Network Measurement. pp. 237–240. PAM'07, Springer-Verlag, Berlin, Heidelberg (2007)
13. Gueye, B., Ziviani, A., Crovella, M., Fdida, S.: Constraint-based geolocation of Internet hosts. IEEE/ACM Trans. Netw. 14(6), 1219–1232 (Dec 2006)
14. Hu, Z., Heidemann, J., Pradkin, Y.: Towards geolocation of millions of IP addresses. In: Proceedings of the 2012 ACM conference on Internet measurement conference. pp. 123–130. IMC '12, ACM, New York, NY, USA (2012)
15. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards IP geolocation using delay and topology measurements. In: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. pp. 71–84. IMC '06, ACM, New York, NY, USA (2006)
16. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for R. Bioinformatics 24(2), 719–720 (2008)
17. Padmanabhan, V.N., Subramanian, L.: An investigation of geographic mapping techniques for Internet hosts. In: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 173–185. SIGCOMM '01, ACM, New York, NY, USA (2001)
18. Poese, I., Uhlig, S., Kaafar, M.A., Donnet, B., Gueye, B.: Ip geolocation databases: unreliable? SIGCOMM Comput. Commun. Rev. 41(2), 53–56 (Apr 2011)
19. Siwpersad, S.S., Gueye, B., Uhlig, S.: Assessing the geographic resolution of exhaustive tabulation for geolocating Internet hosts. In: Proceedings of the 9th international conference on Passive and active network measurement. pp. 11–20. PAM'08, Springer-Verlag, Berlin, Heidelberg (2008)
20. Spring, N., Mahajan, R., Anderson, T.: The causes of path inflation. In: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. pp. 113–124. SIGCOMM '03, ACM, New York, NY, USA (2003)
21. Wang, Y., Burgener, D., Flores, M., Kuzmanovic, A., Huang, C.: Towards street-level client-independent IP geolocation. In: Proceedings of the 8th USENIX conference on Networked systems design and implementation. pp. 27–27. NSDI'11, USENIX Association, Berkeley, CA, USA (2011)

22. Wong, B., Stoyanov, I., Sirer, E.G.: Octant: a comprehensive framework for the geolocalization of Internet hosts. In: Proceedings of the 4th USENIX conference on Networked systems design & implementation. pp. 23–23. NSDI'07, USENIX Association, Berkeley, CA, USA (2007)