

---

**Public Review for**  
**RIPE IPmap Active Geolocation:  
Mechanism and Performance Evaluation**

Ben Du, Massimo Candela, Bradley Huffaker, Alex C. Snoeren, and  
kc claffy

Knowledge about the geographic locations of Internet routers and servers is highly valuable for research on various aspects of Internet structure, performance, economics, and security. Whereas databases for geolocation are commercially available and targeted mostly at end hosts, RIPE offers an open IPmap platform, including its single-radius engine, for geolocation of core Internet infrastructure. This paper introduces the research community to the IPmap single-radius engine and evaluates effectiveness of this method versus commercial geolocation databases NetAcuity and GeoLite2. Access to ground truth constitutes a major challenge in conducting such evaluation studies. The paper collects IP addresses for its study from three sources: virtual machines from the Ring of the Netherlands Network Operators' Group, M-Lab Pods operated by Google, and CAIDA's Ark monitors. The ground-truth dataset is further diversified through addition of IP addresses that are small latency away from Ark monitors. The evaluation considers accuracy, coverage, and consistency of geolocation as well as effectiveness of the single-radius method for different types of autonomous systems. The paper manually analyzes a problematic case where single-radius mistakenly geolocates an IP address of a Budapest-based router to Vienna. Finally, the paper provides recommendations to both users and developers of the single-radius method and discusses limitations of the reported evaluation. The main conclusion is that the IPmap single-radius engine geolocates core Internet infrastructure more accurately than the considered commercial databases and that Internet researchers can greatly benefit from using the IPmap platform for their geolocation needs.

*Public review written by*  
**Sergey Gorinsky**  
*IMDEA, Spain*

# RIPE IPmap Active Geolocation: Mechanism and Performance Evaluation

Ben Du  
UC San Diego  
bendu@ucsd.edu

Massimo Candela  
University of Pisa  
massimo.candela@ing.unipi.it

Bradley Huffaker  
UC San Diego  
bradley@caida.org

Alex C. Snoeren  
UC San Diego  
snoeren@cs.ucsd.edu

kc claffy  
UC San Diego  
kc@caida.org

## ABSTRACT

RIPE IPmap is a multi-engine geolocation platform operated by the RIPE NCC. One of its engines, *single-radius*, uses active geolocation to infer the geographic coordinates of target IP addresses. In this paper, we first introduce the methodology of IPmap’s single-radius engine, then we evaluate its accuracy, coverage, and consistency, and compare its results with commercial geolocation databases. We found that 80.3% of single-radius results have city-level accuracy for our ground truth dataset, and 87.0% have city-level consistency when geolocating different interfaces on the same routers. Single-radius provided geolocation inferences for 78.5% of 26,559 core infrastructure IP addresses from our coverage evaluation dataset. The main contributions of this paper are to introduce and evaluate the IPmap single-radius engine.

## CCS CONCEPTS

• **Networks** → **Network measurement; Network performance analysis.**

## KEYWORDS

Internet Measurement, Active Geolocation, RIPE IPmap

## 1 INTRODUCTION

Geolocation of Internet edge hosts is a service offered by several commercial players, because parties such as online retailers and advertisement companies need locations of existing and potential customers to tailor content or comply with license restrictions. Corporations like Google and Skyhook Wireless have developed systems to collect GPS coordinates from consumer devices [27] and determine customer locations based on their Wi-Fi network [35]. YouTube uses such technology to serve region-specific content or balance load in the face of congestion [34]. In contrast to end hosts, geolocation of core infrastructure such as routers has a smaller market and thus less attention from commercial players. But Internet researchers need to geolocate core infrastructure for many studies, including analysis of interconnection topology and congestion [18], evaluating impacts of regulations [26], and classifying malicious activity [16]. Over the last decade researchers have explored geolocation inference using active measurements.

In 2017, Trammell et al. [45] showed that the accuracy of active geolocation methods such as constraint-based-geolocation (CBG) [22] varies significantly depending on the proximity of vantage points to the targets. Accurate CBG requires vantage points within

sufficiently small round trip time (RTT) from targets, which requires geographic proximity to targets. In 2018, Trammell et al. [46] described such situations as “lucky,” since the worldwide coverage of vantage points determines the probability of finding one close to the target. With over 10K measurement nodes distributed across 179 countries [42], the RIPE Atlas [43] measurement platform provides researchers with by far the best opportunity for this luck.

In 2017, RIPE Atlas introduced an RTT-based module to its IPmap geolocation platform [41], including a public API to access geolocation inferences for specific target IP addresses. This module, called the *single-radius engine*, infers the coordinates of a target IP address using the location of the Atlas probe with the lowest RTT to the target [12]. This methodology seeks out Trammell’s lucky situations, which leads to questions about its performance: How often is it lucky, and how accurate are the results? We approach these questions by evaluating four aspects of single-radius:

- **Accuracy:** We validate single-radius inferences against our ground truth dataset and compare them with commercial databases (NetAcuity [5] and MaxMind [3]).
- **Probe Selection Effectiveness:** we calculate the distance between the selected probe and the target in our ground truth dataset.
- **Coverage:** We examine how many addresses from our interconnection IP dataset single-radius geolocated.
- **Consistency:** We compare single-radius geolocation inferences for different interfaces known to be on the same routers.

## 2 BACKGROUND AND RELATED WORK

Existing IP geolocation techniques fall into two main categories: passive geolocation and active geolocation.

Passive geolocation exploits information such as DNS hostnames, DNS LOC records (RFC1876 [17], rarely populated), crowd-sourced user location reports, resource registry information, and commercial databases. DNS hostname-based geolocation exploits conventions that some ISPs follow to encode geographic information such as airport codes or city names in router hostnames. This approach maps an IP address to a hostname through a reverse DNS lookup, and extracts encoded geographic hints from the hostname. Huffaker et al. [24] explored naming conventions for second-level domains and used active measurements, location dictionaries, and machine learning to extract geographic hints from domain-specific naming conventions. Scheitle et al. [44] mixed passive and active geolocation techniques by extracting a list of candidate cities from hostname substrings and confirming them with ping measurements.

Commercial geolocation datasets provide a static mapping between IP address ranges and geographic locations. The methodology for updating such datasets is proprietary, but likely relies on a combination of public datasets, e-commerce logs, active geolocation, and manually-provided updates by network operators. They often start with public datasets such as the Regional Internet Registries (RIRs) WHOIS databases which provide a contact address for allocated IP address ranges. E-commerce logs enable mapping users' IP addresses to geographic information they provide such as shipping addresses. Operators may also actively submit corrections to such databases. These commercial databases are primarily used for e-commerce, targeted advertising, and meeting geographically scoped legal requirements; thus, they focus on accurately locating end users (edge hosts) in the developed world. As such they often provide low accuracy for IP addresses used on Internet infrastructure such as routers and servers [21, 23, 38].

Active constraint-based geolocation uses probes, usually ping or traceroute, from a set of vantage points with known locations to collect end-to-end delay and IP topological measurements between vantage points and targets. Delays are converted into geographical distances using some distance-delay coefficient that constrains the target's set of possible locations. This set can be further constrained using topology knowledge, such as a shared path with a different IP address whose location is known, or heuristics such as preferentially inferring locations with higher populations.

The simplest approach, introduced by Venkata et al. [37], is to geolocate the target to the location of the vantage point with the lowest latency. A more advanced approach is multilateration, presented by Gueye et al. [22], which geolocates a target to the intersection of distances calculated by multiple points of view, thus establishing a continuous space of answers instead of a discrete one. Other researchers have used information about intermediate hops in traceroutes (RTTs, reverse DNS, and alias resolution) to infer target locations [20, 28, 31].

In many geolocation methodologies, the distance-delay coefficient adopted is derived from  $\frac{2}{3}$  of the speed of light, which is the speed of a signal traversing an optical fiber [28, 45]. Candela et al. [14] suggested that infrastructure diversity leads to distance-delay coefficients that vary by region. For example, it is common for the Middle East to experience inflated RTTs from indirect routing since many interconnection links for this region are in Europe [13]. Hence, Candela et al. divided the world into seven regions and calculated a coefficient for each region by means of latency measurements and a ground-truth dataset.

Most active geolocation techniques require either significant computation or preprocessing. The single-radius engine in the RIPE IPmap platform is designed to be closer to a real-time active geolocation solution, and therefore chooses computational efficiency over more complex measurement approaches.

### 3 SINGLE-RADIUS METHODOLOGY

Single-radius performs four steps for every geolocation request:

- (1) Map the target IP to the AS announcing its containing prefix using RIPE RIS BGP data [39]. Find a set of RIPE Atlas probes topologically close to the target IP (see § 3.1). Schedule a ping measurement from the selected probes. Return an estimated measurement duration to the user.

- (2) Collect all resulting RTTs and discards those above 10ms. Convert remaining RTTs to one-way latencies,  $RTT/2$ . This filtering assumes that geolocation using distant probes (e.g. on another continent) is not effective [45].

- (3) Select probe  $p$  with minimum latency, and convert it to distance  $d$  using a distance-delay coefficient of  $\frac{2}{3}c$ .

- (4) Use location of  $p$  as the center of circle  $C$  with radius  $d$ . Select 100 closest cities to  $p$  using the RIPE Worlds database [9], based on the shortest distance between  $p$  and the city. Select only cities inside circle  $C$ , hence lower latencies yield fewer cities. Finally, rank cities (see § 3.2) and return the highest ranked one to user.

#### 3.1 Initial Probe Selection

Unlike most previous efforts, which used all available vantage points, single-radius limits the number of probes it dedicates to a single geolocation measurement. While using all vantage points provides maximum possible coverage [14], it does not scale in practice. For RIPE Atlas, this approach requires 10,000 ping measurements for each geolocation of a single IP address. Not only would this drastically increase the usage of the Atlas platform and the time needed for measurements to be scheduled and collected, but it would also cause ICMP rate limiting close to the probes [25], and trigger anti-flood and DDoS alerts [47].

The design goal is to use the fewest measurements possible to find probe  $p$  with the lowest RTT  $\pi$  to the target. We would like  $p$  to be a member of the initial set of probes. Single-radius tries to achieve this by selecting probes that are topologically, if not geographically, near the target  $t$  by selecting probes either in  $AS(t)$ , the AS of target, or in AS-level neighbors of  $AS(t)$ . Single-radius creates a list of cities  $C$  and a list of ASes  $A$  as follows.

- (1) Add  $AS(t)$  to  $A$ .
- (2) Add to  $C$  the cities where  $AS(t)$  has a probe.
- (3) Add to  $A$  the ASes neighbors (BGP distance 1) of  $AS(t)$ . Data from [40].
- (4) Add to  $C$  the cities with Internet Exchange Points (IXPs) where  $AS(t)$  is present. Data from PeeringDB [6].
- (5) Add to  $A$  the ASes present at the IXPs identified in step 4.
- (6) Add to  $C$  all the cities corresponding to the facilities where  $AS(t)$  is present. Data from PeeringDB [6].
- (7) Add to  $A$  the ASes peering at facilities identified in step 6.

Once the lists are populated, single-radius places up to 500 probes into the *probe-list*, in the following order: 1) Select up to 100 random probes from  $AS(t)$ ; 2) Select up to 10 random probes from each AS in  $A$ ; 3) Select up to 50 probes for each city in  $C$ . If the engine is not able to infer the inter-domain topology and the  $A$  and  $C$  lists are empty, select 700 random probes worldwide (the effectiveness of this approach is examined in § 6.2).

#### 3.2 Ranking Cities

Single-radius creates a list of at most 100 possible cities, sorted in order of probable location of  $t$ . The rank is weighted by three parameters: 1) the distance of the city from  $p$ ; 2) the number of facilities and IXPs present in the city [6]; 3) the population of the city [9] (similarly to what done by Cicalese et al. [15]). The distance of the city from  $p$  has a weight inversely proportional to the latency (i.e.,  $weight = 10 - \pi$ ). When the returned RTT  $\pi$  is high, the

Cont.	Ark		NLNOG		M-Lab		Ark-prox		Total	
	IP	ASN	IP	ASN	IP	ASN	IP	ASN	IP	ASN
AF	26	25	8	7	7	6	0	0	41	34
AS	26	21	28	20	14	10	90	17	158	57
EU	60	50	359	323	55	16	35	17	509	375
NA	80	59	89	71	69	22	4	2	242	140
OC	9	7	11	11	5	3	56	32	81	46
SA	16	15	11	11	5	5	21	5	53	29

**Table 1: Number of IPs and AS coverage per continent.**

distance from  $p$  has a lower impact in the final rank. The reason behind using the number of facilities and IXPs present in the city as a parameter is that IPmap focuses on geolocation of core Internet Infrastructure.

## 4 DATASETS

### 4.1 Commercial Geolocation Databases

We compare city-level accuracy of single-radius against two commercial databases using data from October 2019: Digital Element’s NetAcuity and MaxMind’s free GeoLite2. NetAcuity claims to be 99.99% accurate on the country-level and 97% on the city-level [5], and MaxMind states its GeoIP2, the paid version has 99.8% country-level accuracy and 86% city-level accuracy within the U.S. in 2019 [33]. We consider these values to be upper bounds on their accuracy. Gharaibeh et al. [21] find that for 4,869 known IP address locations in the RIPE NCC service region, NetAcuity and MaxMind GeoLite2 achieve 90.0% and 70.5% country-level accuracy, respectively. In the ARIN service region, they find NetAcuity and MaxMind are 88.6% and 78.9% country-level accurate, respectively, on 10,711 ground truth addresses.

### 4.2 Ground-Truth Dataset

Our ground-truth dataset is composed of hosts belonging to three measurement platforms: NLNOG Ring Nodes [36], M-Lab Pods [30], Ark Monitors [1]. The location of the hosts is retrievable from each platform’s website. A summary is presented in Table 1.

Netherlands Network Operators’ Group (NLNOG) Ring is a collection of virtual machines on participating organization’s servers. Each organization has its own AS number and is connected to the Default Free Zone. As of October 2019 there were 515 Ring nodes in 55 countries. We include the IP addresses of 500 nodes that were reachable by ping.

Measurement Lab (M-Lab), operated by Google, hosts Pods in data centers and ISPs around the world. As of October 2019, there were 155 pods in 35 countries. We add public IP addresses of 148 M-Lab Pods that were reachable by ping.

CAIDA’s Archipelago (Ark) hosts are deployed in residential, academic, and ISP networks. There were 217 hosts in 54 countries as of October 2019. Many Ark monitors are behind firewalls and only 123 responded to ping; we used the methodology from [21] to geolocate IP addresses from Ark traceroutes in the September 2019 Routed 24 topology dataset [8] within 0.5ms from Ark monitors to the same known location as the Ark monitor. These IPs are likely within 26.5km [14] or at most 33.3km [28] from the Ark monitor. We obtain 206 *ark-proximity* hosts that were reachable by ping,

which increases AS diversity by 55.6% compared to that of the responding Ark monitors alone. After inspection, we remove nine entries from our dataset due to malformed coordinates. Our final dataset has 968 addresses in 651 ASes and 84 countries.

### 4.3 Consistency: CAIDA Topology Dataset

CAIDA’s Macroscopic Internet Topology Data Kit contains IPv4 router-level topology knowledge derived by performing alias resolution on traceroutes to each /24 prefix collected by Ark [7]. We infer routers using two alias resolution techniques, MIDAR and iffnder [11, 29]. We randomly select 540 routers and up to 4 interfaces on each router from 82 million routers in the April 2019 router collection. We refer to this dataset as the *ip-alias* dataset. We use this data set to evaluate whether single-radius geolocates a pair of interfaces on the same router to the same location.

### 4.4 Coverage: MANIC Interconnection Dataset

We use CAIDA’s MANIC dataset to obtain IP addresses of inter-domain links [4]. The interdomain link’s IP addresses are inferred using CAIDA’s border mapping algorithm *bdmap* from the traceroutes collected by CAIDA’s Ark monitors [32]. By choosing a week-long snapshot since September 1, 2019, we obtain 26,559 interconnection IP addresses that represent core Internet router infrastructure. We refer to this dataset as the *manic-links* dataset and use it to evaluate the coverage of single-radius.

## 5 EVALUATION OF SINGLE-RADIUS

We describe our method for evaluating the accuracy, coverage, and consistency of single-radius, as well as how we compare it with NetAcuity and MaxMind GeoLite2. We use the great-circle distance to calculate the geographical distance between any two coordinates, denoted as  $GC(latlong_a, latlong_b)$ .

**Accuracy:** we evaluate the accuracy of single-radius results on our ground truth dataset. We define **error distance**  $d_{err}$  using Equation 1, which calculates the distance between true coordinates  $latlong_{actual}$  and the inferred coordinates obtained from single-radius, NetAcuity, and MaxMind  $latlong_{inferred}$ .

$$d_{err} = GC(latlong_{actual}, latlong_{inferred}) \quad (1)$$

**Probe Selection Effectiveness:** Section 3.1 explains how single-radius attempts to select probes geographically near the target IP address by selecting probes topologically close at the AS level. Since larger ASes have greater geographical and topological diversity, we suspect the single-radius probe selection mechanism will perform poorly on very large ASes. Dhamdhere et al. [19] classifies ASes into four types based their peer degree and customer degree. We adopt this classification scheme with the modification where we only consider the customer degree of an AS, and divide the ASes into three classes to test probe selection performance:

- **Enterprise Customers and Content/Access/Hosting Providers (EC/CAHP):**  $CustomerDegree \leq 2$
- **Small Transit Providers (STP):**  $2 < CustomerDegree \leq 180$
- **Large Transit Providers (LTP):**  $CustomerDegree > 180$

We obtain the customer degree of ASes from ASRank [2]. We quantify probe selection effectiveness using **probe proximity distance**  $d_{ppd}$  defined by Equation 2. We collect from the RIPE Atlas

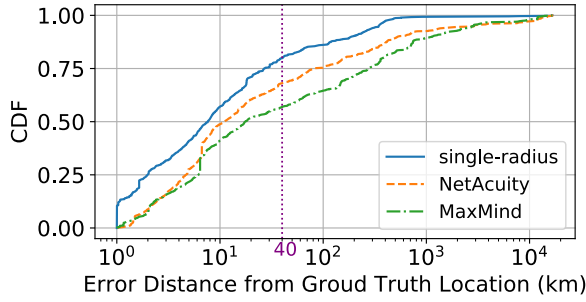


Figure 1: Geolocation results vs. ground truth error.

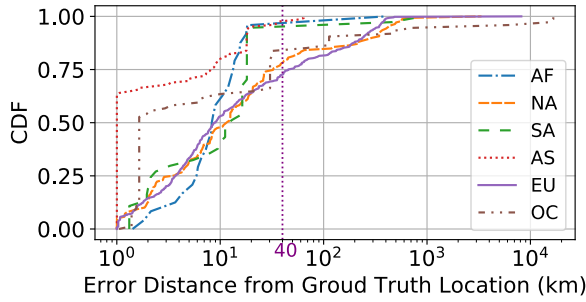


Figure 2: Regional breakdown of single-radius accuracy.

platform the coordinates  $L_{closest}$  of the Atlas probe closest to the target IP, and  $L_{selected}$  coordinates of Atlas probe selected by single-radius probe selection.

$$d_{ppd} = GC(L_{target}, L_{selected}) - GC(L_{target}, L_{closest}) \quad (2)$$

We also examine the geolocation error using Equation 1 for IP addresses in each category.

**Coverage:** we count the number of *manic-links* IP addresses that single-radius geolocated given a 3-day window and compute its percentage of successfully geolocated IP addresses. We use our ground truth data to explore the relationship between geolocation accuracy and maximum RTT.

**Consistency:** we evaluate the consistency of single-radius results on our *ip-alias* dataset. All interfaces of a router should geolocate to the same location. After obtaining the geographic coordinates  $latlong_1, latlong_2, \dots, latlong_n$  of  $n$  interfaces  $I_1, I_2, \dots, I_n$  on each router, we calculate the **router-level disagreement distance**  $d_{RLD}$  using Equation 3:

$$d_{RLD} = \max_{i,j} GC(latlong_i, latlong_j) \quad (3)$$

This equation first calculates the great circle distances between the inferred geolocations of every pair of interfaces on the same router, and then takes the maximum for each router as the **router-level disagreement distance**.

## 6 RESULTS

### 6.1 Accuracy

We queried all 968 IP addresses from our ground truth dataset and obtained 870 results, against which we evaluate the accuracy of

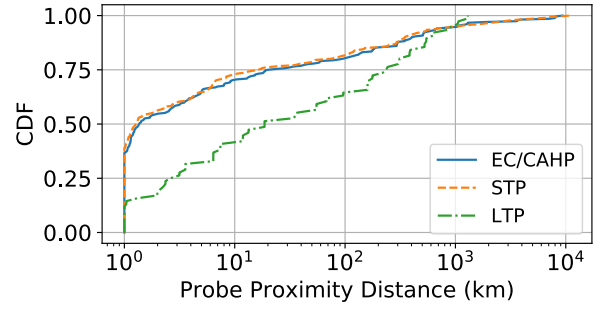


Figure 3: Probe proximity distances are larger for large transit providers.

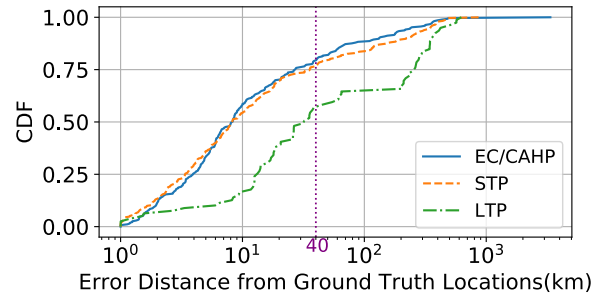


Figure 4: Error distances are greater for large transit providers.

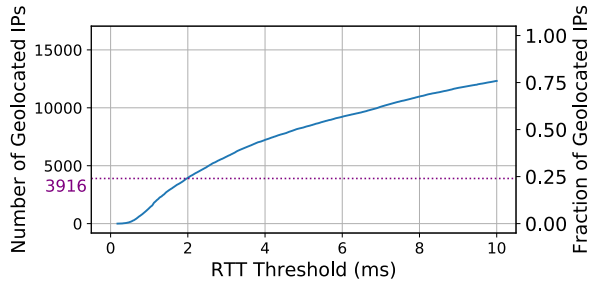
single-radius. We consider geolocations within 40 km of the true locations to be accurate, a threshold used in previous geolocation studies [21] because it reflects a metropolitan area perimeter.

Figure 1 shows the error distance of single-radius, NetAcuity, and MaxMind. Single-radius outperformed the other two services on our ground truth dataset. Single-radius achieved median, 75th, and 95th percentile error distances of 6, 26, and 344 km, respectively. In contrast, those of NetAcuity and MaxMind were 10, 80, and 2867 km, and 17, 278, and 2886 km, respectively. Over 80% of single-radius results were within the 40-km error threshold (vertical purple line).

Furthermore, the accuracy of single-radius was higher in Africa, Asia, Oceania, and South America than in North America and Europe (Figure 2). In the first four continents, most ground truth IP addresses mapped to a few well-connected cities, which single-radius geolocated easily (Table 2).

Continent	AS	AF	SA	OC	NA	EU
Accuracy	97.6%	96.0%	94.7%	84.0%	78.3%	73.3%
# of cities	34	27	18	13	100	178

Table 2: Single-radius had higher accuracy (error < 40 km, top row) when ground truth IP addresses (bottom row) were concentrated in fewer cities.



**Figure 5: Coverage of single-radius at different RTT thresholds. Single-radius geolocates 75.8% of MANIC interconnection addresses using 10 ms as RTT threshold. With a threshold of 2 ms, only 25% of the IPs are geolocated.**

## 6.2 Probe Selection Effectiveness

To study how single-radius probe selection performs for the AS categories defined in § 5, we downloaded 647 measurements tagged with single-radius and active-geolocation from RIPE Atlas and classified their target IPs: 274 IPs in EC/CAHP ASes, 292 IPs in STP ASes, and 81 IPs in LTP ASes. Figure 3 shows the median probe proximity distance for EC/CAHP, STP, and LTP ASes are 0.36 km, 0.27 km, and 17.63 km respectively. Median probe proximity distances in LTP ASes are two orders of magnitude greater than that of EC/CAHP and STP ASes, which indicates that single-radius’ probe selection mechanism is less effective for larger ASes.

To further illustrate the impact of probe selection on geolocation accuracy, Figure 4 shows the distribution of error distances from locations of IP addresses in our ground-truth dataset for each AS category. We find that 80.6%, 76.1%, and 55.8% of single-radius results are within the 40-km error threshold for EC/CAHP, STP, and LTP ASes. That is, IPs classified in LTP ASes exhibit larger error rates across for all sources.

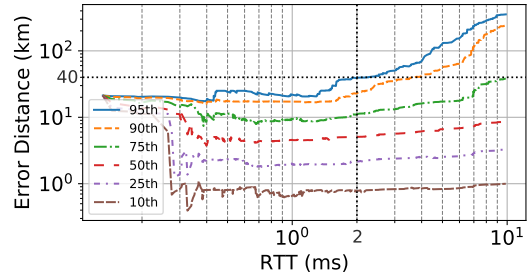
## 6.3 Coverage

Of the 26,559 *manic-link* IP addresses, only 16,245 responded to any RIPE Atlas ping. We kept only these addresses in our coverage dataset and excluded those that were unreachable at the time of the experiment. Single-radius provided results for 12,319 (78.5%) of the reachable addresses. Figure 5 shows the relationship between coverage and RTT threshold. Switching from the default 10 ms threshold to 5 ms drastically dropped coverage to 51.1%, but increased accuracy.

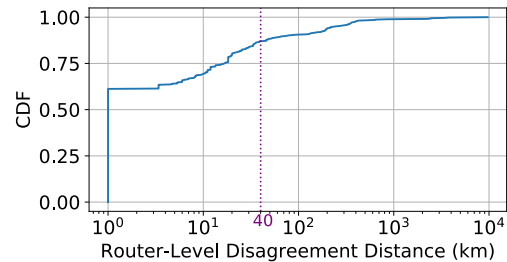
Figure 6 shows the relationship between accuracy and RTT threshold using our ground truth dataset. The error distances were almost all below 40 km. We see the 95-th percentile of error distances exceeds 40 km when the maximum RTT exceeds 2 ms. Were a user to choose a 2-ms threshold, the *manic-link* coverage would further decrease to 3,916 (24.1%) according to Figure 5.

## 6.4 Router-Level Consistency

Figure 7 plots the router-level disagreement distances of 540 routers from § 4.3 according to Equation 3 in § 5. The plot shows the median and 75-th percentile router-level disagreement distances were 0 km and 16 km respectively. Using 40 km again as the city-level consistency threshold, we found 470 out of 540 (87.0% of) geolocated



**Figure 6: A 2-ms threshold keeps the error distance < 40 km.**



**Figure 7: 61.3% of routers had all interfaces geolocated by single-radius to the exact same coordinates.**

routers with city-level consistency. Next we analyze a case in which the inferred geolocation of the interfaces on the same router were more than 40km away from each other.

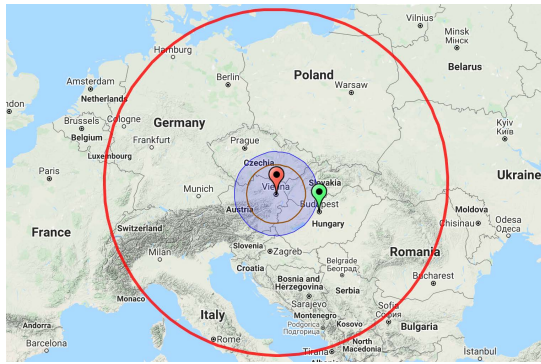
## 6.5 Case Study: Inconsistent Geolocation

We examine an interesting case: a single router with two IP addresses, one of which single-radius geolocated to Vienna, Austria and the other single-radius geolocated to Budapest, Hungary. Using traceroutes and reverse DNS lookup, we manually confirmed these two addresses were from two interfaces on the same router located in Budapest. We analyzed single-radius’ procedure (§ 3). Figure 8 illustrates the conflict that led to incorrect geolocation of one router interface to Vienna, which followed these steps:

1. In the compiled *prob-list*, the probe with the lowest RTT, 7.86 ms, was in Vienna (red marker). We call this Atlas probe  $P_{\text{vienna}}$ .
2. Single-radius translated the RTT into geographical distance  $d$  using  $\frac{2}{3}c$  as distance-delay coefficient. The red bold circle with radius  $d$  and center of  $P_{\text{vienna}}$  shows the inferred possible geographic region of the target.
3. RIPE IPmap selected 100 cities inside the red bold circle closest to  $P_{\text{vienna}}$ . The blue shaded region circumscribes the geographic scope of the selected 100 cities.
4. Finally, the ranking module returned Vienna as the highest among all 100 cities as it had the most population.

We identify two factors that contribute to this erroneous geolocation. First and foremost, single-radius probe selection fails to select an Atlas probe in Budapest for one router interface during its final random selection process (§ 3.1), and instead, selects  $P_{\text{vienna}}$ . Second, Budapest is not within the blue shaded circle centered around Vienna (Figure 8), hence not considered during ranking. Single-radius’ selection of only the closest-100 cities may not be effective in areas with numerous small cities.





**Figure 8: Single-radius geolocated two interfaces of the same router to separate cities. The engine computed the red circle centered around the red marker, and used the blue shaded circle to circumscribe the 100 cities closest to the red marker.**

## 7 RECOMMENDATIONS

Our experience yields many insights that suggest potential improvements to single-radius, although some require a deeper analysis of trade-offs in scalability (including load on the RIPE Atlas platform), reliability, and immediacy.

**For users:** We recommend that users make a conscious tradeoff between accuracy and coverage, using Figures 5 and 6. Users must separately obtain the geolocation at [https://ipmap.ripe.net/api/v1/locate/\[IP\]/partials/single-radius](https://ipmap.ripe.net/api/v1/locate/[IP]/partials/single-radius) and RTT from selected Atlas probes to target at [https://ipmap.ripe.net/api/v1/single-radius/\[IP\]](https://ipmap.ripe.net/api/v1/single-radius/[IP]). Users can then choose to accept or reject the geolocation result based on the corresponding RTT.

**For RIPE IPmap developers:** 1) Using  $\frac{2}{3}c$  as the distance-delay coefficient ignores RTT inflation caused by factors such as queuing delay and routing circuitousness, and likely overestimates geographic distance traveled by packets. We recommend customizing this coefficient per region as described by Candela et al. [14] We experimented with changing the value of  $\frac{2}{3}c$  in § 3 step (3) to region-specific distance-delay coefficients, and repeated subsequent steps for the same IPs used for the accuracy analysis in § 6. This change improved (dropped) the geolocation error in only 0.96% of the cases; in no cases did the geolocation error increase. The modest impact of the regional distance-delay coefficient is due to the fact that single-radius nominates at most 100 cities closest to the selected Atlas probe within the red circle. For example, in Figure 8, the red bold circle is drawn using a distance-delay coefficient of  $\frac{2}{3}c$ , the smaller brown circle is drawn using a regional coefficient, and the shaded blue area covers approximately the closest 100 cities. The highest ranked city, used in our evaluation, is usually so close to the probe returning the minimum RTT that the tighter constraint introduced by the regional coefficient made no difference. However, this tighter constraint significantly reduced the number of returned cities for a target, by 60.53% on average, thus reducing result ambiguity. In no case did the tighter constraint erroneously exclude the true city of the target. Such tighter geographical constraints would work better with a multilateration approach.

2) Single-radius uses only the result provided by the single probe with the lowest RTT to the target. A multilateration engine that uses

results from multiple Atlas probes would allow IPmap to increase coverage in regions where Atlas node deployment is sparse.

3) Removing intrinsic RTT inflation due to connection types (e.g. DSL, satellite links) may improve accuracy. Bajpai et al. [10] show last-mile latency on different residential network providers remains stable over time. Since single-radius might select Atlas probes on residential networks, subtracting last-mile latencies from measured RTTs may significantly improve accuracy.

4) Using the minimum of multiple RTTs collected over a longer period would filter out noise from temporarily inflated latencies.

5) For cases where the ping RTT is high (e.g. greater than 5 ms), single-radius could initiate a traceroute to the target, gathering path information to help identify convoluted routing.

6) Single-radius could tune the number of cities close to the probe based on the size of the metropolitan area of the target (§ 6.5).

## 8 LIMITATIONS

**Ground truth inadequacy.** Due to a small ground-truth dataset, our accuracy evaluation is not comprehensive. IP geolocation ground-truth data is hard to obtain. To avoid bias, we cannot consider RIPE Atlas probes in the ground-truth dataset, since RIPE IPmap uses RIPE Atlas for the measurements.

**Geographical bias of ground truth.** Most IP addresses from our ground-truth dataset are located in Western Europe and the contiguous U.S. Single-radius accuracy appears to differ by region, so the results shown here may not hold in all regions.

**AS sampling bias.** Most of our ground truth IP addresses belong to EC/CAHP and STP ASes, which leads to an imbalanced number of data points when analyzing the effectiveness of single-radius' probe selection mechanism.

**Limited access to RIPE NCC platforms.** We only test some of the recommendations for improving IPmap because certain experiments require deploying system-wide changes to the platform. We are collaborating with RIPE NCC for future experiments.

## 9 SUMMARY

We described in detail the methodology of RIPE IPmap's single-radius engine, and evaluated its accuracy, coverage, and consistency. We evaluated its accuracy on 870 ground-truth addresses and compared results against two commercial geolocation databases. We find 80.3% of the inferred geolocations have city-level accuracy, and higher accuracy than both commercial databases. We find that single-radius' median error distance is 17.27-km greater on larger ASes compared to smaller ASes, consistent with the fact that for larger ASes (which tend to have wide geographic footprints), single-radius' probe selection mechanism is more likely to select an Atlas probe farther away from the target IP. We evaluated its coverage on 16,245 core router infrastructure (interconnection) IP addresses from September 2019 and find that single-radius produces results on 78.5% of them. We evaluated its consistency on multiple interfaces of 540 routers from the April 2019 Internet Router Topology dataset [7], and find that single-radius achieves city-level consistency on 87.0% routers. We believe the RIPE IPmap single-radius engine, and the IPmap platform more generally, can contribute significantly to Internet geolocation research.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) Award C-ACCEL OIA-1937165 and Award OAC-1724853. We would also like to appreciate the RIPE NCC for their collaboration.

## REFERENCES

- [1] 2019. Archipelago (Ark) Measurement Infrastructure. <https://www.caida.org/projects/ark/>
- [2] 2019. AS Rank. <https://asrank.caida.org/>
- [3] 2019. MaxMind: IP Geolocation and Online Fraud Prevention. <https://www.maxmind.com>
- [4] 2019. Measurement and Analysis of Internet Congestion. <https://manic.caida.org/>
- [5] 2019. NetAcuity Industry-Standard Geolocation - Digital Element. <https://www.digitalelement.com/geolocation>
- [6] 2019. PeeringDB: The Interconnection Database. <https://www.peeringdb.com/>
- [7] 2019. The CAIDA Internet Topology Data Kit. <https://www.caida.org/data/internet-topology-data-kit/>
- [8] 2019. The CAIDA UCSD IPv4 Routed /24 Topology Dataset-September 2019. <https://doi.org/10.23721/107/1354084>
- [9] 2019. Worlds API. <https://ipmap.ripe.net/static/doc/#/?id=worlds-api>
- [10] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. 2017. Dissecting Last-mile Latency Characteristics. *SIGCOMM Comput. Commun. Rev.* 47, 5 (Oct. 2017), 25–34.
- [11] CAIDA. 2019. iffinder. <https://www.caida.org/tools/measurement/iffinder/>
- [12] Massimo Candela. 2019. RIPE IPmap - What's Under the Hood? [https://labs.ripe.net/Members/massimo\\_candela/ripe-ipmap-whats-under-the-hood](https://labs.ripe.net/Members/massimo_candela/ripe-ipmap-whats-under-the-hood)
- [13] M. Candela, E. Gregori, V. Luconi, and A. Vecchio. 2019. Dissecting the Speed-of-Internet of Middle East. In *2019 Proceedings IEEE INFOCOM Workshops*. 720–725.
- [14] Massimo Candela, Enrico Gregori, Valerio Luconi, and Alessio Vecchio. 2019. Using RIPE Atlas for Geolocating IP Infrastructure. *IEEE Access* 7 (2019), 48816–48829.
- [15] Danilo Cicalese, Diana Joumblatt, Dario Rossi, Marc-Olivier Buob, Jordan Auge, and Timur Freeman. 2015. A Fistful of Pings: Accurate and Lightweight Anycast Enumeration and Geolocation. <https://perso.telecom-paristech.fr/drossi/paper/rossi15infocom.pdf>. In *IEEE INFOCOM*.
- [16] David Dagon, Guofei Gu, Christopher P Lee, and Wenke Lee. 2007. A taxonomy of botnet structures. In *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*. IEEE, 325–339.
- [17] Christopher Davis, Ian Dickinson, Tim Goodwin, and Paul Vixie. 1996. *A means for expressing location information in the domain name system*. RFC 1876. RFC Editor. <http://www.rfc-editor.org/rfc/rfc1876.txt>
- [18] Amogh Dhamdhare, David D. Clark, Alexander Gamero-Garrido, Matthew Luckie, Ricky K. P. Mok, Gautam Akiwate, Kabir Gogia, Vaibhav Bajpai, Alex C. Snoeren, and Kc Claffy. 2018. Inferring Persistent Interdomain Congestion. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. ACM.
- [19] A. Dhamdhare and C. Dovrolis. 2011. Twelve Years in the Evolution of the Internet Ecosystem. *IEEE/ACM Transactions on Networking* 19, 5 (Oct 2011), 1420–1433.
- [20] Brian Eriksson, Paul Barford, Bruce Maggs, and Robert Nowak. 2012. Posit: a lightweight approach for IP geolocation. *ACM SIGMETRICS Performance Evaluation Review* 40, 2 (2012), 2–11.
- [21] Manaf Gharaibeh, Anant Shah, Bradley Huffaker, Han Zhang, Roya Ensafi, and Christos Papadopoulos. 2017. A Look at Router Geolocation in Public and Commercial Databases. In *Internet Measurement Conference (IMC '17)*. ACM.
- [22] Bamba Gueye, Artur Ziviani, Mark Crovella, and Serge Fdida. 2006. Constraint-based Geolocation of Internet Hosts. *IEEE/ACM Trans. Netw.* 14, 6 (Dec. 2006), 1219–1232.
- [23] B. Huffaker, M. Fomenkov, and k. claffy. 2011. *Geocompare: a comparison of public and commercial geolocation databases*. Technical Report. Center for Applied Internet Data Analysis (CAIDA).
- [24] Bradley Huffaker, Marina Fomenkov, and kc claffy. 2014. DRoP: DNS-based Router Positioning. *SIGCOMM Comput. Commun. Rev.* 44, 3 (July 2014), 5–13.
- [25] Mattia Iodice, Massimo Candela, and Giuseppe Di Battista. 2019. Periodic Path Changes in RIPE Atlas. *IEEE Access* 7 (2019), 65518–65526.
- [26] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolaos Laoutaris. 2018. Tracing Cross Border Web Tracking. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. ACM, 14. <http://doi.acm.org/10.1145/3278532.3278561>
- [27] Russel Kipp Jones, Farshid Alizadeh-Shabdz, Edward James Morgan, and Michael George Shean. 2008. Server for updating location beacon database. US Patent 7,414,988.
- [28] Ethan Katz-Bassett, John P. John, Arvind Krishnamurthy, David Wetherall, Thomas Anderson, and Yatin Chawathe. 2006. Towards IP Geolocation Using Delay and Topology Measurements. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC '06)*. ACM, 71–84. <http://doi.acm.org/10.1145/1177080.1177090>
- [29] K. Keys, Y. Hyun, M. Luckie, and k. claffy. 2013. Internet-Scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Transactions on Networking* 21, 2 (Apr 2013).
- [30] Measurement Lab. 2019. M-Lab. <https://www.measurementlab.net/status/>
- [31] Sándor Laki, Péter Mátray, Péter Hága, Tamás Sebök, István Csabai, and Gábor Vattay. 2011. Spotter: A model based active geolocation service. In *2011 Proceedings IEEE INFOCOM*. IEEE, 3173–3181.
- [32] M. Luckie, A. Dhamdhare, B. Huffaker, D. Clark, and k. claffy. 2016. bdrmap: Inference of Borders Between IP Networks. In *ACM Internet Measurement Conference*.
- [33] MaxMind Support Center. 2019. How accurate are your GeoIP2 and GeoIP Legacy databases? [https://support.maxmind.com/geoip-faq/geoip2-and-geoip-legacy-databases/](https://support.maxmind.com/geoip-faq/geoip2-and-geoip-legacy-databases/how-accurate-are-your-geoip2-and-geoip-legacy-databases/)
- [34] R. Mok, V. Bajpai, A. Dhamdhare, and k. claffy. 2018. Revealing the Load-balancing Behavior of YouTube Traffic on Interdomain Links. In *Passive and Active Measurement Conference (PAM)*.
- [35] Uday Nagendran. 2004. Methods of using wireless geolocation to customize content and delivery of information to wireless communication devices. US Patent 6,731,940.
- [36] NLNOG. 2019. NLNOG Ring. <https://ring.nlnog.net/>
- [37] Venkata N Padmanabhan and Lakshminarayanan Subramanian. 2001. An investigation of geographic mapping techniques for Internet hosts. *ACM SIGCOMM Computer Communication Review (CCR)* 31, 4 (2001), 173–185.
- [38] Ingmar Poese, Steve Uhlig, Mohamed Ali Kaafar, Benoit Donnet, and Bamba Gueye. 2011. IP geolocation databases: Unreliable? *ACM SIGCOMM Computer Communication Review* 41, 2 (2011), 53–56.
- [39] RIPE NCC. [n.d.]. Routing Information Service (RIS). <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris>
- [40] RIPE NCC. 2010. RIPEstat. <https://stat.ripe.net/>
- [41] RIPE NCC. 2018. RIPE IPmap. <https://ipmap.ripe.net>
- [42] RIPE NCC. 2019. Global RIPE Atlas Network Coverage. <https://atlas.ripe.net/results/maps/network-coverage>
- [43] RIPE NCC staff. 2015. RIPE Atlas: A Global Internet Measurement Network. *Internet Protocol Journal* 18, 3 (2015), 2–26.
- [44] Quirin Scheitle, Oliver Gasser, Patrick Sattler, and Georg Carle. 2017. HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks. *CoRR* abs/1706.09331 (2017). arXiv:1706.09331 <http://arxiv.org/abs/1706.09331>
- [45] Brian Trammell. 2017. On the Suitability of RTT Measurements for Geolocation. <https://github.com/britram/trilateration/blob/master/paper.ipynb>
- [46] Brian Trammell and Mirja Kühlewind. 2018. Revisiting the Privacy Implications of Two-Way Internet Latency Data. In *Passive and Active Measurement*, Robert Beverly, Georgios Smaragdakis, and Anja Feldmann (Eds.).
- [47] J Udhayan and R Anitha. 2009. Demystifying and rate limiting ICMP hosted DoS/DDoS flooding attacks with attack productivity analysis. In *2009 IEEE International Advance Computing Conference*. IEEE, 558–564.

## A REPRODUCIBILITY

All the datasets used in this research are of public access. We provide here the complete list for easy access:

- RIPE IPmap single-radius engine: <https://ipmap.ripe.net/api/v1/single-radius/>
- CAIDA Ark Monitors: <https://www.caida.org/projects/ark/locations/>
- NLNOG Ring Nodes: <https://api.ring.nlnog.net/1.0/nodes/active>
- M-Lab Pods: <https://siteinfo.mlab-oti.measurementlab.net/v1/sites/geo.json>
- CAIDA Routed IPv4 /24 Topology Traceroute (*ark-proximity* dataset): [https://www.caida.org/data/active/ipv4\\_routed\\_24\\_topology\\_dataset.xml](https://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml)
- CAIDA Macroscopic Internet Topology Data Kit (ITDK): <https://www.caida.org/data/internet-topology-data-kit/>
- CAIDA MANIC API: <https://api.manic.caida.org/v1/>
- CAIDA AS Rank: <https://api.asrank.caida.org/v2/docs>