Computer Science
Technical Report

Delay-based Identification of Internet Block Movement

Manaf Gharaibeh
Colorado State University

Christos Papadopoulos
Colorado State University

John Heidemann
University of Southern California/ISI

Craig Partridge
Colorado State University

April 7, 2020

Colorado State University Technical Report CS-20-101

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

Phone: (970) 491-5792 Fax: (970) 491-2466
WWW: https://compsci.colostate.edu

Delay-based Identification of Internet Block Movement

Manaf Gharaibeh[1], Christos Papadopoulos[1], John Heidemann[2], and Craig Partridge[1]

[1] Colorado State University
[2] University of Southern California/Information Sciences Institute

Abstract. Some IP blocks occasionally change their physical location, such as when blocks are transferred to different organizations, or repurposed within an organization. IP geolocation systems need to identify such changes to provide accurate results for *location-dependent* applications such as geo-blocking and online fraud prevention. We propose an *efficient method to identify IP blocks that move*, since full geolocation is expensive and unnecessary for blocks that do not move. Our approach uses persistent changes in latency as an indicator of block movement, tracking all ping-responsive IPv4 /24 blocks from a handful of globally distributed vantage points. We estimate around 2.1% of the 3.77M /24 blocks we studied have changed location at least once in the last 3 months of 2018. We find that the remaining blocks were consistently *RTT-stable* during the same period, suggesting that their locations were also stable. We validate a random sample of blocks we identify as moving and confirm 80% (41 of 51) through traceroutes.

1 Introduction

IP Geolocation systems report the physical location of an IP address. Geolocation is widely available as a commercial service (e.g., [7, 8, 20, 25]), and several approaches have been studied in the academic literature (e.g., [5,17,21,36,37]). Regardless of the geolocation approach used, the result only provides a snapshot of the current IP-to-location mappings. Some IP blocks occasionally move to a different location, for example, when transferred to a different organization, or reassigned within an organization. As a result, the previously estimated locations become outdated and need to be updated.

The geolocation accuracy can have a significant impact on Internet applications that utilize geolocation information. For example, major Video on Demand (VoD) services use geo-blocking to limit or block access to their content based on users' location [11,26]. These services are popular. Futuresource Consulting reports that Subscription Video on Demand (SVoD) reached 60% of households in North America, 26% in Western Europe, 21% in Asia-Pacific, and 19% in Latin America, with a revenue of more than $29 billion in 2018 [13]. Digital TV Research reports that Gross SVoD subscriptions increased to 508 million in 2018 [9]. With such mammoth revenues and subscriber populations, geolocation error can lead to significant loss of revenue and many dissatisfied customers. This paper looks for Internet block movement as a sign that the current geolocation information needs to be updated.

To preserve the freshness of IP-to-location mappings, a geolocation system needs to identify when a block moves, and then update its location. One may obtain information about block movement from the Regional Internet Registries (RIRs) transfer reports, which may report IP address range transfers between organizations [3]. However, the reported date of transfer does not necessarily reflect when the block actually appears in a new location (§5.5). More importantly, these reports do not include information about ISPs reassignment of blocks to other locations. *Measurement-based* geolocation methods can maintain up-to-date geolocation if applied continuously, but these methods can be intrusive and inefficient when applied continuously to the entire IP address space.

The primary goal of this chapter is to define a method to identify when blocks (/24 IPv4 prefixes) move in order to help a geolocation system to maintain up-to-date IP-to-location mappings. The identification of a block movement tells a geolocation system it is time to re-run geolocation to update the block location. To achieve this goal, we propose a *delay-based* method that monitors ping measurement to visible /24 blocks from 6 globally distributed vantage points. We show that these measurements can identify block movement and are inexpensive enough to run continuously. Our method identifies movement by observing persistent changes in the latency state of a /24 block from multiple sites around the same time.

Previous work on IP geolocation focused on defining methods to identify the location of IP addresses [5,6,24,30,36], and assessing the accuracy of public and commercial geolocation services [15,19,27,31]. Our work is different from both categories; we do not define a new IP geolocation method. Instead, we propose a lightweight method to *identify if a block has moved* from one location to another, signaling the need to re-run an existing geolocation algorithm. Unlike delay-based geolocation methods, our method works well with only a handful of vantage points regardless of their distance from the targets. We do not map latency estimates to location constraints. Instead, we use them as fingerprints to dictate location stability. We are not aware of previous work that focuses on identification of IP block movement.

The first contribution of this paper is defining an efficient method to identify IP block movement from delay measurements observed via a small number of vantage points. Our second contribution is the application of our method to a dataset of 3.77M /24 blocks, showing that 2.1% of them experienced movement during the last 3 months of 2018.

2 Datasets

Our work uses a USC ICMP echo-request (ping) data coverage to look for block movement in about 4M /24 blocks of the responsive Internet (§2.1). We then use two datasets from CAIDA to validate a sample of our block movement findings: the IPv4 Routed /24 Topology Dataset (§2.2), and the Internet Topology Data Kit (§2.3).

2.1 Latency Information from the USC Internet Outage Data

Our method evaluates delay measurement to /24 IP blocks over time, so we require Internet-wide data that contains frequent latency estimates. While many groups today conduct censuses of the IPv4 address space, such scans are often infrequent or lack latency estimates. We instead extract latency estimates from publicly available measurements taken for Internet outage detection [33] using Trinocular [28]. This data is available to researchers at no cost; we obtained it from USC.

Trinocular scans IP addresses in about 4 million responsive IPv4 /24 network blocks using ICMP echo-request messages. (The target list of blocks is updated periodically using long-term history data from Internet censuses [12].) Each /24 block is probed every 11 minutes, one or more probes taking place, often stopping after the first successful probe returns an echo-reply. Each block is therefore probed about 130 times a day. Successful replies include the round-trip time; we ignore unsuccessful probes. Scans rotate through different addresses in each block over time.

Trinocular collects data from six vantage points (VPs) positioned around the world. We use data from all of the six vantage points collected during October through December of 2018 [34]. The VPs identifiers and locations are: **c** (center of the U.S in Ft. Collins, Colorado), **e** (east coast of the U.S. in Arlington, Virginia), **g** (Athens, Greece), **j** (Tokyo, Japan), **n** (Utrecht, Netherlands), and **w** (west coast of the U.S. in Marina del Rey, California).

There are about 130 measurements attempts per day for each block; we estimate block latency each day from these observations (§3.1). Each attempt tries up to 15 addresses and reports latency only if one replies. We determine a block's latency state status only for days with 3 or more VPs each have 10 or more latency observations, which we refer to as *determination-valid* days. About 156k (3.9% of all blocks in the ping dataset) do not have any determination-valid days and around 41k (1%) have 9 or less such days. In the remainder of this paper, we use the remaining 3.77M blocks (95.1% of all blocks) with 10 or more determination-valid days.

2.2 Paths from the CAIDA UCSD IPv4 Routed /24 Topology Dataset

To examine the relationship between observed changes in latency estimates and routing changes, we use the *CAIDA UCSD IPv4 Routed /24 Topology Dataset* (henceforth referred to as *CAIDA-topology* dataset in this paper) [2] We use historical traceroute data from the same period as our ping dataset (§2.1).

The CAIDA-topology traceroute measurements are collected using around 152 *Ark monitors*, globally distributed in 52 countries. These monitors work as a team to probe randomly selected IP addresses in every routed /24 prefix. Only a single random destination in a /24 prefix is probed every 48 hours by only one of the monitors.

To get observations from locations near our six vantage points, we first identified active Ark monitors that are within 50 km of our probes. We found 15 such Ark monitors close to 4 of our vantage points. We used the closest Ark monitors available for the other two vantage points; the Ark monitor *wbu-us* at Boulder, CO for the VP_c at Fort Collins, CO, U.S., and *sof-bg* at Sofia, Bulgaria for the VP_g at Athens, Greece.

Ideally, we would like to compare one monitor's traceroute measurements soon before and after a block sees a change in latency. We first identify blocks with changes in their latency according to §3. For a given change, we select one monitor's measurements if they satisfy the following criteria: First, the measurements should be from the *consistent-RTT* periods (i.e., periods with no other changes detected in them.). before and after the change. Second, only monitors with at least one measurement before the delay change, and another after it are used. If a monitor has multiple measurements that satisfy these two criteria, we select the closest in time to the observed change.

We do not always find relevant traceroute measurements that satisfy our criteria. Each routed IPv4 /24 prefix in the CAIDA-topology dataset is probed only once every 2 days, by only one of the Ark monitors, which may or may not be one of the 17 Ark monitors we selected. Applying the selection methodology to blocks we identify with delay changes shows that around 62% of the changes have relevant traceroute measurement.

2.3 Paths from the CAIDA Internet Topology Data Kit

For a more comprehensive routing path comparisons, we augment the measurements of the CAIDA-topology dataset (§2.2) with data from *CAIDA's Macroscopic Internet Topology Data Kit* (ITDK) [1]. We use the ITDK data to map any traceroute hop IP address to its router-level node, AS, and location. In this paper, we use the ITDK 2018-03 dataset, the closest public ITDK dataset in 2018 to our ping dataset.

The ITDK data has two router-level topologies, derived with different alias resolution tools. We used the one derived with MIDAR [23] and iffinder [22] tools, which CAIDA reports as the more accurate, although with less coverage. The dataset extracts the IP addresses of intermediate hops that appear in traceroute measurement performed using the Ark infrastructure.

AS assignments are derived using RIPE and RouteViews BGP tables and Regional Internet Registries (RIR) delegations. The geolocation is derived at the router-level using different sources that include hostname mapping, information from known Internet eXchange (IX) point, and MaxMind's free GeoLite City database. A router is assigned a location only when all of its identified interfaces are individually geolocated to the same location.

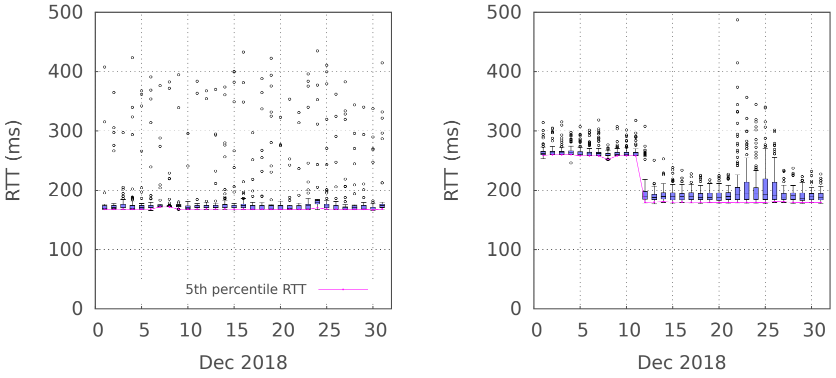
3 Methodology: From Block Latency to Block Movement

Our methodology begins by processing observations of block RTT to get a stable estimate of its latency (§3.1). We then show examples of the patterns in these estimates that indicate block movement, congestion, and routing changes (§3.2). Our detection method searches for these patterns to detect block movement (§3.3).

3.1 Stable Estimation of VP-to-Block Latency

To get a stable estimate of block latency we must filter through measurement noise due to network congestion, route changes, and other transient network effects. We use *5%ile of all daily RTT observations* (or just *5%ile-RTT*) as our stable estimator of the block's latency. This near-minimum estimate of RTT filters out queuing delay from congestion while avoiding outliers.

Figure 1 depicts the box plots of two sample /24 blocks daily RTT observations and their 5%ile (the magenta line). Each plot depicts daily observations for one VP over one month. Each box shows the interquartile range (IQR), with RTT observations with the lower and upper quartiles (25%ile and 75%ile) with an interior line showing the median. The lower (and upper) whiskers show the lowest RTT still within (1.5× IQR) of the lower (or upper) quartile, and circles show outliers beyond the whiskers.



(a) VP *w* and block 129.16.71.0/24.

(b) VP *n* and block 45.226.48.0/24.

Fig. 1: Box plots and 5%ile RTT for sample /24 blocks. Dataset: 2018q4.

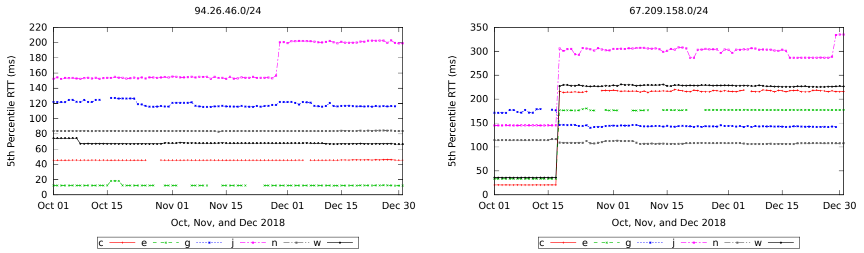
First, we observe that quartiles are quite tight, suggesting that we can filter outliers. Moreover, most outliers are above the upper whiskers, suggesting that the 5%ile will be close from minimum RTT and therefore speed-of-the-Internet latency. Having established this estimator, we report only the 5%ile-RTT in later sections and omit quartiles and outliers.

Second, we can see that these two VPs show different latency fingerprints. For Figure 1a, while RTT observations vary some throughout the day, the daily 5%ile is relatively stable over time. In §5, we show that this *5%ile-RTT stability* is common for most blocks. By contrast, the block in Figure 1b also shows 5%ile-RTT stability, but with *two modes*, values before and after December 12 each show a different common value, and latency drops by about 78 ms on the 12th. This change indicates either routing or location change, and in §3 we describe our algorithm to identify blocks movement by detecting such changes across *multiple* VPs.

3.2 Common Patterns in IP-Block Latency

For each day, the 5%ile-RTTs from 6 VPs defines the that block's *latency state* (or just *block latency*). We look for patterns of change in block latency that indicate block movement. Here we show sample patterns drawn from blocks in 2018q4. We use the insights from these examples to define our algorithm in the next section.

Figure 2 depicts the 5%ile-RTTs for two sample /24 blocks, where each line represents the data from one vantage point. For the block in Figure 2a, the daily observations are mostly consistent up to November 29 for all 6 VPs. This RTT-stability suggests the block's location is fixed during that period. This pattern is common in most blocks in our ping dataset, as we show in §5. We expect that a change in a block location would affect the observations of multiple VPs. Only VP_j observed a significant change on November 29, suggesting an event that affected only that VP and not the block (e.g., a routing



(a) Latency change in one VP suggests a routing change. (b) Latency change in all VPs suggests block movement.

Fig. 2: Example latency states for blocks showing routing change (left) and movement (right). Dataset: 2018q4.

change on the path from VP_j to that block). We also see other, less significant changes in delay, including VP_n on October 8, VP_e on October 16, and VP_g on several occasions. Many small changes at different times suggest routing changes or persistent congestion, not block movement. Our algorithm in §3 looks for persistent, significant latency change to indicate movement and filter out other effects.

Figure 2b shows an example of a /24 block with a significant change in the block latency observed by multiple vantage points on October 18. Some of the vantage points experienced an RTT increase (VP_c, VP_e, VP_j, and VP_w), while VP_g observed a decrease, and VP_n observed an insignificant change. We also note that the new block latency after October 18 is persistent through the end of December. This pattern indicates that the block likely moved to a different location on November 18, 2018.

3.3 Identifying Block Movement from Latency Measurements

Building on 5%ile-RTT estimates from 6 VPs, and common patterns to look for, we now present *our algorithm for block movement*.

Our algorithm have four steps, each confirming the block is suitable for continued analysis. First, we determine blocks with sufficient latency observations as *determination-valid*, then we look for *changes in VP latency*, that show *VP agreement*. Our final check is *persistence* of the change. We review each of these steps below.

Our first step is to confirm that the block has *determination-valid* days. A block's one day worth of measurement is determination-valid when, over the course of that day, it has enough VPs, each with enough observations that we can draw statistically strong conclusions. We require that each VP have at least 10 latency estimates, so that that VP's 5%ile is valid (not mislead by transient network conditions). We require observations from at least 3 VPs, so that we can confirm movement and not just a route change affecting one path. Since each VP makes about 130 attempts to measure latency and we have 6 VPs, these requirements (of 10 estimates per VP and 3 VPs) allow for substantial downtime or measurement error.

Our next step is to look for *changes in VP-to-block latency*. In §3.2, we showed stationary blocks see some variation in daily 5%ile-RTTs. We consider latency from a VP to the block to have a *significant change* if the change exceeds some threshold T

compared to the long-term average (one week). We use a threshold of a 9% change in latency, as determined from ROC analysis from training data (§4.3).

We use *VP agreement* to filter out routing changes. Physical movement usually affects all VPs, but changes in internet paths will change latency between a VP and the target block, and are unlikely to affect all VPs. We consider an *agreed-latency-change* to occur when there are VP-block latency changes by at least half of the VPs (3 or more). We do not demand the agreement of all VPs as some might not have VP-latency-valid days around the time a change happens. Moreover, some VPs may see insignificant changes concerning the delay-change threshold criteria.

Finally, we require that the latency change is *persistent*. IP blocks are unlikely to move frequently or for short periods. Therefore, latency changes caused by a physical block movement will likely persist more than just a few days. We compute the duration of an agreed-latency-change as the number of days until we observe another change, or until the VPs agreement heuristic is broken (i.e., we no longer have 3 or more VPs with significantly different block latency to that before the agreed-latency-change). In this work, we focus on agreed-latency-changes that persist for at least one week as a strong indication of block movement.

When a block meets these four criteria we consider it a *movement candidate*.

4 Controlled Experiments with Synthetic Data

To evaluate our method with known ground truth, we next describe how we simulate block movement (§4.1), build a test dataset from synthetic movement of real observations (§4.2), and use this to select optimal parameters for our method (§4.3).

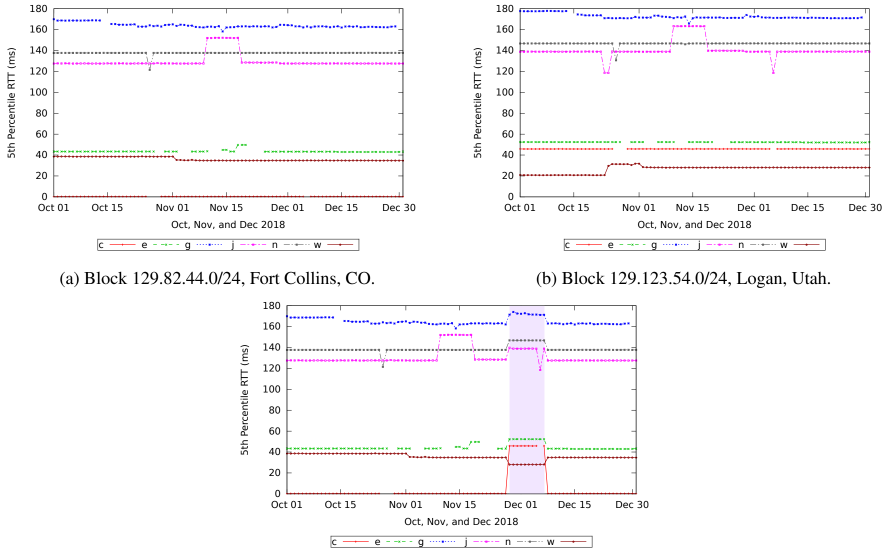
4.1 Simulation of Block Movement

We simulate block movement by replacing a block's RTT data in a selected range of days with data from another block at a different location. (We use only latency, the addresses involved are unimportant.) We select start and end randomly, with end at least 7 days after start.

Figure 3a and Figure 3b show 5%ile-RTTs of blocks about 700 km apart (in Fort Collins, Colorado and Logan, Utah). They show some variation in latency over the observation month, but neither is identified as moving by our algorithm. In Figure 3c, we create a *synthetic* block that moves from Colorado (the base data) to Utah from November 28 to December 7 (the shaded region) by replacing that period of data.

4.2 Building a Dataset with Synthetic Movement

Next, we build a dataset with synthetic movement of block. We begin with 60 /24 IP blocks, each with websites from a different university. We verify that each block appears to be physically at its university (and not outsourced to a third party) using *WHOIS* information, and reverse DNS names, checking that the name indicates the university. We select university blocks because academic institutions have known locations and often self-host, suggesting their blocks are at static, known locations.



(a) Block 129.82.44.0/24, Fort Collins, CO.

(b) Block 129.123.54.0/24, Logan, Utah.

(c) A synthetic /24 block with movement in the shaded period.

Fig. 3: Real blocks (dataset: 2018q4) combined to make a synthetic block with known movement (dataset: synthetic).

Table 1: Accuracy from method to the university blocks. Dataset: 2018q4.

Delay-change threshold	3%	5%	7%	9%	11%	13%	15%
False Positives	13	4	3	2	1	0	0
True Negatives	47	56	57	58	59	60	60

We select geographically distributed blocks: 8 universities in Africa, 9 in Latin America, 10 in Asia, 13 in Europe, and 20 in North America. We use the *GeoNames geographical database* [14], to identify the geographic location of each block from its university.

Since our goal is to identify blocks that move, we consider all 60 university blocks as *negative instances* of this category. We verify the blocks do not cause false positives in our algorithm by applying movement identification with *delay-change* thresholds from 3% and 15% (with 2% increments). Table 1 shows false positives (FP) and true negatives (TN) per delay-change threshold. We drop a delay-change threshold of 3% as too sensitive, since with it, 13 blocks (21.7%) are misclassified as moving. Larger thresholds show a few or no false positives. We identify 4 blocks that are false positives at higher thresholds, so we eliminate them from use in synthesis.

Using the remaining, verified 56 blocks, we create all 1,540 possible pairs where we insert data from one block into another to simulate movement (as in Figure 3). Most of the pairs (85%) show uniform distances from 0 to 12000 km. (Distribution is in §A.)

Table 2: Accuracy for different delay-change thresholds from known movement. Dataset: synthetic.

Threshold	TPs	FNs vs distance (km)					
		any distance (1540)	0-100 (7)	101-200 (11)	201-300 (12)	301-400 (8)	401-500 (19)
5%	1516	24	5	5	5	1	2
7%	1505	35	5	6	5	1	3
9%	1495	45	5	6	7	2	4
11%	1474	66	5	7	8	4	8
13%	1449	91	5	8	11	5	11
15%	1434	106	5	8	11	5	14

4.3 ROC Analysis

We now use this synthetic dataset of 1,540 blocks with known movement (§4.2) to select optimal parameters for our algorithms with ROC-curve (Receiver Operating Characteristic) analysis.

Table 2 shows how many blocks move (TPs) or do not have detected movement (the FNs at any distance) *vs.* different delay-change thresholds. The table also compares FNs against distance ranges below 500 km, since short distance movement is more challenging. The number in parentheses under each range is the count of instances we have in that range. Geographically closer blocks show smaller differences in latency, and our method performs better as distances increase. Smaller thresholds seem to have fewer false negatives since they are more sensitive.

We use the ROC-curves to select the delay-change threshold that balances true positive rate (TPR) with false positive rate (FPR). We compute the TPR and FPR at various thresholds from the confusion matrix of the analysis of university blocks (Table 1) and the synthetic-moving blocks Table 2. Figure 4 shows the ROC curve for the TPR against FPR at different thresholds (shown next to the marks on the graph). We see that a threshold of 9% yields good TPR against FPR results (97% and 3.3%), allowing for detecting most of the moving blocks while reducing false positives over the evaluation datasets. We use this threshold in the remainder of this paper.

We next test the sensitivity of our method and the selected delay-change threshold (9%) on a more challenging subset of the synthetic blocks. Rather than using all possible 1,540 synthetic combinations, we perform the ROC-curves on synthetic pairs of distance less than 4,500 km (comparable to the horizontal width of the African continent or a large country such as the U.S.). We find 23% of the synthetic blocks (360) that satisfy this criteria. As expected, the results show lower TPRs but are still consistent with those in Figure 4, showing that our method still identify most of the moving blocks (§B). The threshold 9% still yields good TPR against FPR (91% and 3.3%), confirming our results over all synthetic blocks.

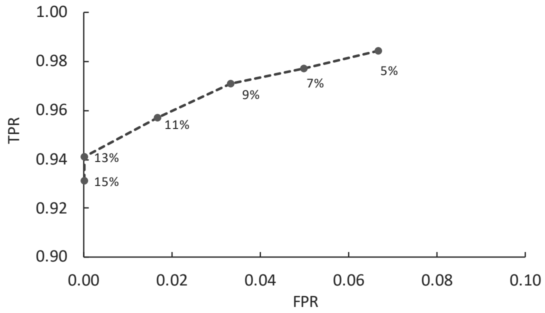


Fig. 4: ROC curve showing true and false positive rates over blocks with known movement. Annotates next to points show the threshold, and the *y*-axis does not start at zero. Dataset: synthetic.

Table 3: Block movement from Internet-wide data. Dataset: 2018q4: threshold: 9%.

ping-dataset blocks	˜4M	
determination-valid Blocks	3.77M	100%
consistent *5%ile-RTT*	3.33M	88.3%
agreed-latency-change	441k	
short	362k	9.6%
indicate movement	78.7k	2.1%

5 Evaluation with Real-World Data

Having defined our methodology with controlled experiments, we now apply our method to real-world data for about 3.77M /24 blocks to understand the Internet (§5.1) and verify real-world block movement (§5.4 and §5.5).

5.1 Applying Our Method in the Wild

We next apply our method to Internet-wide data (§2.1) to identify blocks that move. Our goal is to identify how many blocks move during 2018q4. We do not expect many blocks to move, since most organizations have a fixed physical location, and assignment of addresses is often stable.

We summarize our results in Table 3, using our method with a delay-change threshold of 9%. We see that 78.7k (2.1% of the 3.77M /24 we consider) move at least once during the last 3 months in 2018. These results show that our algorithm can identify a subset of moving blocks that IP geolocation services should review, saving 98% of the effort of checking everything.

Most of the blocks in our ping dataset showed consistent 5%ile-RTT. About 3.33M (88.3%) show consistent latencies (no agreed-latency-change) over the in 3 months, meaning at no time there was a significant latency change agreed upon by 3 or more VPs. Another 362k blocks (9.6%) see one or more short agreed-latency-changes, but

Table 4: Block movement from Internet-wide data. Dataset: 2019q1: threshold: 9%.

Ping dataset blocks	4M	
determination-valid Blocks	3.82M	100%
Consistent *5%ile-RTT*	3.49M	91.3%
agreed-latency-change	333k	
Short	268k	7.2%
Indicate movement	65k	1.7%

Table 5: Frequency of a block movement. Dataset: 2018q4.

Changes	**Count**	**% of total**
1	62,289	79.2%
2	13,665	17.4%
3	2,095	2.7%
4	569	0.7%
5	46	0.1%
6	14	0.0%
total	78,678	100%

return quickly. These short changes suggest transient network events such as routing change or congestion, not movement.

5.2 Movement over Time

To compare movement rates over time, we apply our block movement identification algorithm to the data of a different quarter, the first quarter of 2019 [35]. Table 4 shows the results. The total number of /24 blocks probed in 2019q1 is just a few hundreds less than 4M. Around 3.82M of these blocks are determination-valid blocks (§2.1).

We identify about 65k (1.7% of the determination-valid blocks) to have moved during the first 3 months of 2019, around 14k fewer blocks compared to the 2018q4 results (Table 3). Overall, the results over the two quarters are consistent. We again observe that most of the blocks are RTT-stable and identify a small fraction of the responsive blocks as moving.

As future work, we plan to extend our study of block movement using data from additional quarters. This longitudinal study can help quantify the rate at which blocks move over time, showing which blocks are location-stable and which are more dynamic.

5.3 Frequency of Movement

To identify if there are blocks that move more frequently than others, we show next the frequency of movement for the 78.7k blocks we identify as moving. Table 5 shows 6 categories of exclusive location-changes we find, ranging from 1 to 6. We see that the majority of blocks identified as moving experienced only one location change (79%). We find about 17% with 2 changes. Less than 3.5% experienced between 3 and 6 changes.

Table 6: Validation of block movement with traceroute data. Datasets: 2018q4, ITDK, and CAIDA-topology.

candidates	79k	
random samples	100	
no traceroutes	40	
with traceroutes	60	
misses AS-criteria	9	
passes AS-criteria	51	100%
no or int. change	10	20%
near end change	41	80%
country change	26	
city change	8	
no city data	7	

Table 7: Validation of block movement with transferred blocks. Datasets: 2018q4 and RIR reports.

ARIN reported transfers	2,416	
no latency	2,400	
have latency	16	
lack before or after	13	
lack before	5	
lack after	8	
have before-and-after	3	100%
confirmed move	2	67%
did not move	1	33%

5.4 Validating Block Movement with Historic Traceroutes

Next, we use the CAIDA-topology (§2.2) and ITDK (§2.3) datasets to confirm our block movement findings with router-level path information.

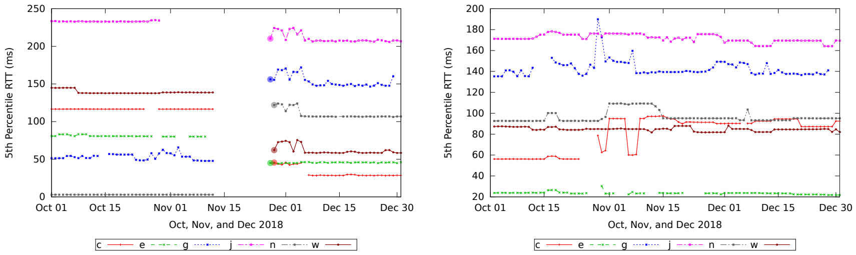
We start by finding relevant traceroute data for a moving block, as described in §2.2. We then use the ITDK dataset to map traceroute hops (interfaces) to routers (for de-aliasing), ASes, and locations (§2.3). We require the AS of the last identified hop to match the AS of the target block (the *AS-criteria* for traceroutes).

We consider two traceroute hops to match if they map to the same router. Comparing traceroutes from before and after a latency change can show: (a) near-identical routing paths, indicating no movement; (b) change in intermediate routers only, suggesting the latency change is due to a routing change; (c) change towards the target block, suggesting a block has moved. We confirm (c) when the penultimate hop changes AS.

We evaluate 100 randomly chosen blocks from the 79k that we identify as moving, and show the results in Table 6. Of those 100 blocks, 60 have traceroutes, and 51 traceroute to the target AS and so can be used to test work. We find 10 of these 51 have near-identical traceroutes or show intermediate routing changes—these blocks are likely false positives due to congestion or routing changes. The remaining 41 blocks show traceroutes with different penultimate ASes, suggesting movement—about 80% show true positives. Geolocation confirms 26 (of 41) map to changed country and 8 changed city, confirming movement. The remaining 7 changed AS but we could not confirm movement because they lacked city-level geolocation.

5.5 Validating Block Movement with Transferred Blocks

We next examine blocks between Internet regions (defined by RIRs, the Regional Internet Registries: ARIN, RIPE, APNIC, LACNIC, AfriNIC). We examine the 74 IP ranges (2,416 /24s) that ARIN reported as inter-RIR transfers in 2018q4 [3]. We have latency data for only 16 /24s in that set, probably because transfers are often of previously unused blocks [4, 29].



(a) Block 185.169.108/24.

(b) Block 69.94.100/24.

Fig. 5: Latency data for two inter-RIR transferred blocks. Dataset: 2018q4.

Out of the 16 blocks, 13 cease responding to ICMP midway through the quarter, suggesting they were transferred but have not yet resumed service. Although we expect these blocks are in the process of moving, our algorithm cannot detect movement until they resume service, so we do not consider them further. There are many such blocks (130k in the quarter); such blocks deserve monitoring for when they resume service and we should reevaluate geolocation.

Examining the remaining 3 blocks: our method identifies 2 as moving. Figure 5a shows the 5%ile-RTTs for one of them, block 185.169.108/24. (The other block is 185.169.109/24 and shows similar results. Both are from BGP prefix 185.169.108/22 and are announced by AS395855.) The 185.169.108/24 block was transferred from an organization in the Netherlands (under RIPE) to a recipient in the U.S. (under ARIN). This block responds to ICMP through November 12, goes silent, and then resumes service November 27. According to ARIN's report, the transfer was effective on October 25, 2018. After the block resumes, observed latency does not stabilize until Dec. 5. We hypothesize that the new block operators were debugging routing over the first week of December. The 5%RTTs before and after the gap are quite different. The data is consistent with this block moving to a new location, as found by our algorithm. It is surprising, though, that it was responsive after the transfer date; perhaps the paperwork preceded routing changes.

The third block is 69.94.100/24, with 5%-ile RTTs shown in Figure 5b. This block was transferred between two different cloud hosting services, but one under ARIN and the other APNIC. Our algorithm do not show it moved, we see some fluctuation in latency around the reported transfer date (November 1), but not persistent changes for most of VPs, and no interruption in service. We hypothesize that this block was transferred with hardware in a data center, so although the RIR responsible for the address space changed, we believe the block did not move. Examination of the exact set of ping responses shows the block behaved identically for the entire three months, consistent with this hypothesis (see §C).

This section complements our prior validation with traceroutes (§5.4) with validation with documented change of allocation. Although we have before-and-after data for only 3 blocks, our data demonstrates movement in two cases and suggests non-movement in the other.

6 Related Work

There is a great deal of prior work on IP geolocation. Much of the prior work on IP geolocation focuses on improving geolocation accuracy [10, 17, 21, 36, 37], usually using delay-based measurement. We do not propose a new geolocation method, but instead show how to harvest existing latency data to detect movement and trigger re-application of existing geolocation.

Other work has shown accuracy depends on VP proximity to the target [18, 36]. Hu *et al.* use a preliminary scan to identify the best vantage points to use [18]; we instead use a lightweight scan to identify blocks to re-geolocate.

Other IP geolocation work studied the accuracy and granularity of public and commercial databases. Siwpersad *et al.* studied the geographic resolution of geolocation databases [32]. They compared public datasets with Constraint-Based Geolocation (CBG) [17]. Gueye *et al.* also used CBG to estimate the max distance between block endpoints to estimate its geographic span [16]. These studies are usually one-time comparison of algorithms; our work instead shows how to do lightweight scans of the entire space to trigger reevaluation of specific blocks for possible movement.

7 Conclusions

We have shown an efficient method that identifies *movement of IP blocks* using existing ICMP scans, based on changes in the latency “fingerprint” from multiple, distributed observers. We validate our approach by confirming movement through traceroutes and information about Internet registration re-allocations. We show that about 2.1% of Internet blocks move over the course of a quarter, and suggest our approach will help IP geolocation providers keep their data up-to-date.

Acknowledgments

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-18-2-0280. and by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via contract number 70RSAT18CB0000014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

1. The CAIDA UCSD Internet Topology Data Kit - march 2018. `http://www.caida.org/data/internet-topology-data-kit`.
2. The CAIDA UCSD IPv4 Routed /24 Topology Dataset - october-december 2018. `https://www.impactcybertrust.org/`.
3. ARIN. Statistics: Specified transfers of internet number resources. `https://account.arin.net/public/transfer-log`, 2019.

4. ARIN. Transferring IP addresses & ASNs. https://www.arin.net/resources/registry/transfers/, October 2019.
5. M. Candela, E. Gregori, V. Luconi, and A. Vecchio. Using RIPE Atlas for geolocating IP infrastructure. *IEEE Access*, 7:48816–48829, 2019.
6. O. Dan, V. Parikh, and B. D. Davison. Improving ip geolocation using query logs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 347–356, New York, NY, USA, 2016. ACM.
7. DB-IP. The DB-IP database. https://db-ip.com, September 2019.
8. Digital Envoy. Digital Element NetAcuity databases. https://www.digitalelement.com/geolocation/, September 2019.
9. Digital TV Research. SVoD databook. https://www.digitaltvresearch.com/ugc/press/254.pdf, 2019.
10. B. Eriksson, P. Barford, B. Maggs, and R. Nowak. Posit: a lightweight approach for IP geolocation. *SIGMETRICS Perform. Eval. Rev.*, 40(2), Oct. 2012.
11. European Commission. A digital single market strategy for europe. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52015DC0192, 2015.
12. X. Fan and J. Heidemann. Selecting representative IP addresses for Internet topology studies. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, New York, NY, USA, 2010. ACM.
13. Futuresource Consulting Ltd. SVoD market research, analysis and commentary. https://www.futuresource-consulting.com/press-release/media-entertainment-press/new-services-set-to-drive-svod-revenues-up-25-to-usd-36-billion-in-2019/, 2019.
14. GeoNames. The GeoNames Geographical Database. http://www.geonames.org/, September 2019.
15. M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, and C. Papadopoulos. A look at router geolocation in public and commercial databases. In *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, pages 463–469, New York, NY, USA, 2017. ACM.
16. B. Gueye, S. Uhlig, and S. Fdida. Investigating the imprecision of IP block-based geolocation. In *Proceedings of the 8th International Conference on Passive and Active Network Measurement*, PAM'07, Berlin, Heidelberg, 2007. Springer-Verlag.
17. B. Gueye, A. Ziviani, M. Crovella, and S. Fdida. Constraint-based geolocation of Internet hosts. *IEEE/ACM Trans. Netw.*, 14(6), Dec. 2006.
18. Z. Hu, J. Heidemann, and Y. Pradkin. Towards geolocation of millions of IP addresses. In *The 2012 ACM conference on Internet measurement conference*, IMC '12, 2012.
19. B. Huffaker, M. Fomenkov, and k. claffy. Geocompare: a comparison of public and commercial geolocation databases - Technical Report . Technical report, Cooperative Association for Internet Data Analysis (CAIDA), May 2011.
20. IP2Location. IP2Location Databases. http://www.ip2location.com, September 2019.
21. E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe. Towards IP geolocation using delay and topology measurements. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006.
22. K. Keys. Iffinder. http://www.caida.org/tools/measurement/iffinder/, 2018.
23. K. Keys, Y. Hyun, M. Luckie, and k. claffy. Internet-Scale IPv4 Alias Resolution with MIDAR. *IEEE/ACM Transactions on Networking*, 21(2):383–399, Apr 2013.
24. Y. Lee, H. Park, and Y. Lee. Ip geolocation with a crowd-sourcing broadband performance tool. *SIGCOMM Comput. Commun. Rev.*, 46(1), Jan. 2016.
25. MaxMind Inc. Maxmind geoip2 city. https://www.maxmind.com/en/geoip2-databases, September 2019.
26. G. Mazziotti. Is geo-blocking a real cause for concern in Europe? *Retrieved from Cadmus, European University Institute Research Repository*, 2015.

27. I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. Ip geolocation databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2), Apr. 2011.
28. L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding internet reliability through adaptive probing. In *Proceedings of the ACM SIGCOMM Conference*, Hong Kong, China, Aug. 2013. ACM.
29. P. Richter, M. Allman, R. Bush, and V. Paxson. A primer on IPv4 scarcity. *SIGCOMM Comput. Commun. Rev.*, 45(2):21–31, Apr. 2015.
30. Q. Scheitle, O. Gasser, P. Sattler, and G. Carle. HLOC: Hints-Based Geolocation Leveraging Multiple Measurement Frameworks. In *Network Traffic Measurement and Analysis Conference (TMA)*, Dublin, Ireland, June 2017.
31. Y. Shavitt and N. Zilberman. A geolocation databases study. *IEEE Journal on Selected Areas in Communications*, 29(10), 2011.
32. S. S. Siwpersad, B. Gueye, and S. Uhlig. Assessing the geographic resolution of exhaustive tabulation for geolocating internet hosts. In *Proceedings of the 9th International Conference on Passive and Active Network Measurement*, PAM'08, Berlin, Heidelberg, 2008. Springer-Verlag.
33. USC/LANDER Project. Internet outage measurements. `https://ant.isi.edu/datasets/outage/`, September 2017.
34. USC/LANDER Project. Internet outage measurements. IMPACT ID `USC-LANDER/internet_outage_adaptive_a34c-20181001`, Oct. 2018.
35. USC/LANDER Project. Internet outage measurements. IMPACT ID `USC-LANDER/internet_outage_adaptive_a35c-20190101`, Jan. 2019.
36. Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang. Towards street-level client-independent ip geolocation. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, Berkeley, CA, USA, 2011. USENIX Association.
37. B. Wong, I. Stoyanov, and E. G. Sirer. Octant: A comprehensive framework for the geolocalization of Internet hosts. In *The 4th USENIX conference on Networked systems design & implementation*, 2007.

A Distribution of Distances in the Synthetic Dataset

Figure 6 shows the distribution of distances of the 1,540 block pairs in the synthetic dataset, described in §4.2.

B Continent-Scale ROC Analysis

Our ROC-curves analysis in §4.3 used all 1,540 synthetic blocks, including ones made with block pairs at different continents. In this section, we test our method and perform the ROC-curves analysis on synthetic blocks that simulate movement within a continent like Africa or large countries such as the U.S. and China. From the 1,540 synthetic blocks (§4.2) we only include those that simulate movement within a distance of 4,500 km. We find 360 (23%) synthetic blocks that satisfy this criteria.

Table 8 shows the true positives (TPs) and false negatives (FNs) of applying our method to the 360 synthetic blocks with known movement within 4,500 km. The results are consistent with those in §4.3 for all 1,540 synthetic blocks, showing that smaller thresholds are better at detecting blocks with smaller movement.

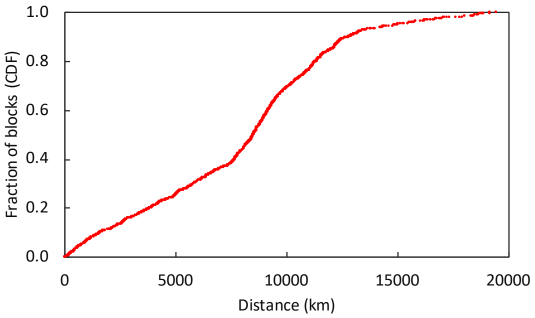


Fig. 6: Distribution of distances in all 1,540 block pairs with known movement. Dataset: synthetic.

Table 8: Accuracy for different delay-change thresholds from known movement at *continent scale*. Dataset: synthetic.

Threshold	TPs	FNs
5%	342	18
7%	335	25
9%	328	32
11%	305	55
13%	291	69
15%	281	79

For the ROC-curves analysis, we compute the TPR from the confusion matrix of synthetic-moving blocks analysis (Table 8), and the FPR from the previous analysis of university blocks (Table 1). Figure 7 shows the ROC curve for the TPR against FPR at different thresholds (shown next to the marks on the graph). Although the movement scale of the 360 synthetic blocks is far smaller than that of all 1,540 blocks, our method still achieves good results. We see that a threshold of 9% again yields good TPR against FPR results (91% and 3.3%) for movement within a continent scale.

C Transferred Block Without Movement

In §5.5 we examined three blocks that were transferred between RIRs, and found that one of them did not appear to move based on our latency observations. To confirm that the block responded identically before and after the transfer date, Figure 8 shows the raw ICMP responses we observed. In the figure, each green dot is a positive response, black are non-responses, and white are addresses that are not probed.

We see that two addresses (last octets 33 and 35) replied consistently through the entire three months, including times both before and after the RIR transfer. Two other addresses (last octets 1 and 3) stopped on 2018-10-12. Continuous addresses are consistent with the block changing administrative responsibility, but *not* actually moving.

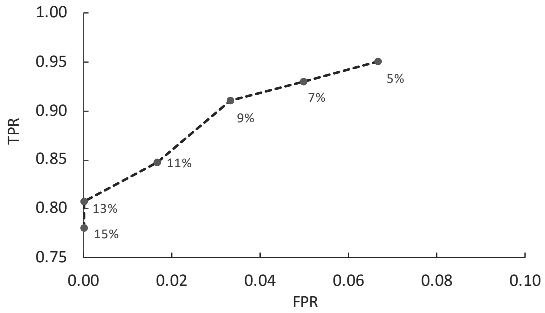


Fig. 7: ROC curve showing true and false positive rates over blocks with known movement within a continent scale. Dataset: synthetic.

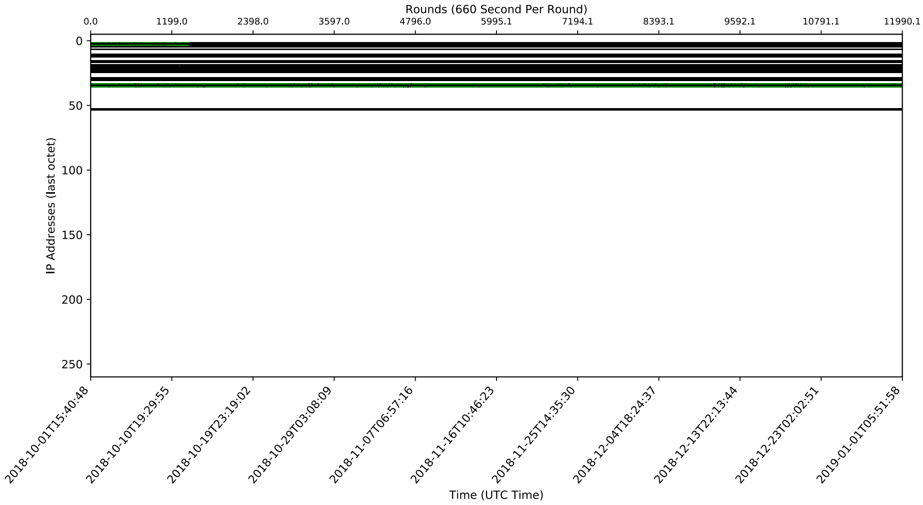


Fig. 8: ICMP responses for block 69.94.100/24, showing similar behavior over all three months.