Turning Off GPS is Not Enough: Cellular location leaks over the Internet

Hamed Soroush, Keen Sung,
Erik Learned-Miller, Brian Neil Levine, and Marc Liberatore

Dept. of Computer Science, University of Massachusetts, Amherst, USA
{soroush,ksung,elm,brian,liberato}@cs.umass.edu

Abstract. Many third parties desire to discover and disclose your location with the help of your cell phone. Using an embedded GPS, phone software will commonly reveal coordinates to carriers, advertisers, and applications. Can a remote party determine locational information absent explicit GPS information? For example, given a known starting or ending point, can a streaming music server distinguish the path you've taken through the physical world? We show that the path a cell phone and its owner take from or to a known location can be determined from remote observations of changes in TCP throughput. Empirically, our method can correctly determine with greater than 78% accuracy the path taken by phone from one of four paths, and with 63% accuracy the path taken from among eight paths.

1 Introduction

Information that is not part of the content of electronic communication, such as the location of the user, has almost no privacy protection in the U.S. Carriers inherently know to which towers mobile phones associate, and investigators can easily compel the release of such information. Mobile phones also expose location information to Internet services, apps [4], and advertising networks [9]. Location privacy in phones is limited to GPS access control per-app. In this paper, we demonstrate that mobile phones are subject to *Internet-based remote localization.* We show that the broad geographic path a cell phone and its owner take can be determined from remotely observed changes in TCP throughput. In short, we show that turning off GPS is not enough.

Many factors shape network traffic between a phone and a remote server. Some are ephemeral, including competing link- or TCP-layer flows. Others change only infrequently, such as the cellular infrastructure and the device's network stack. Signal strength is likely to have some geographical consistency, stemming from the location of radio towers, buildings, and the terrain; any user that lives on the edge of network coverage knows places where a call is likely to drop. A remote party communicating with a phone has a window into the complex interactions of these features. They can be used to reveal the phone's location, or at least significantly narrow the list of possible paths taken by a phone and its owner. This information is leaked regardless of application-level privacy settings.

In this paper, we examine a restricted sub-problem of the general Internet-based remote localization problem. We approach the problem as an instance of the *time-sequence multi-class classification problem* [13], using network traffic traces as instances to be classified, and geographic paths as classes. We compare the accuracy of several classifiers that use only throughput measurements of music streamed to a mobile phone on a 3G/UMTS network. In our subproblem, we assume the attacker knows either the starting or ending location of the user's mobility, can isolate traffic belonging to the user, and can collect labeled training traces of phones traveling along a limited set of possible paths.

Consider a streaming audio service (like that provided by Spotify or Pandora) that wishes to localize users that have disabled GPS. For the service to discover an end point, it geolocates the user when they are connected to a broadband-supported Wi-Fi base station. Most mobile phones in the U.S. preferentially make these connections when they are available. To acquire training data, the service leverages their observations of users who do not disable GPS: travels within a cellular network provide training for throughput distributions over paths; connections from geographically static Wi-Fi provide the service with GPS coordinates of such access points.

Contributions. We collected hundreds of traces of music that we streamed to phones along four geographically separate routes in two directions each. We find that within small geographical areas, mean throughput is largely consistent and distinct. We examine the accuracy of three remote localization classifiers that leverage this consistency. Even a naive approach, trained on the mean throughput of each path, has some success. We trained an HMM classifier on the distribution of throughput values and achieved higher accuracy. Our best performing approach, a k-nearest neighbors (k-NN) classifier, trains on the ordered sequence of throughput values of each route. Empirically, the k-NN can correctly determine with greater than 78% accuracy the path taken by phone from one of four paths, and with 63% accuracy the path taken from among eight paths.

In sum, our main contributions are as follows.

- We define the Internet-based remote localization problem and demonstrate for the first time that cellular phones are subject to limited forms of it.
- We recorded 286 traces of music streamed over a 3G/UMTS cellular connection over one of four geographically distinct, bi-directional routes (8 paths total) over a one-month period. We also recorded 29 stationary traces.
- Our analysis shows statistically different throughput and signal strength means among small geographic areas (0.9 km^2 each). Phones that move between locations travel through consistent and distinct network conditions that are remotely observable.
- We examine the performance of the three classifiers listed above, demonstrating that both the k-NN and HMM approaches can distinguish between a small number of geographic routes taken by mobile users using only throughput measurements. The classifiers are also able to distinguish between mobile and

stationary users. Only the k-NN is able to distinguish between two traces of the same route in opposite directions.

We explain classifiers performance by examining received signal strength at the target, its correlation with throughput, and the geographic consistency of both. We conjecture that the usual defenses against inference attacks, such as traffic padding and shaping, will help defend against this attack.

2 Problem Statement and Attacker Model

We are interested in a subset of the Internet-based remote localization problem: *Can an attacker, providing an Internet-based service to a mobile user that disabled GPS, infer the path taken by the mobile user, from among a limited set of paths, using only information visible at the server?* In later sections of this paper, we show that the answer is yes. Here, we elaborate on the problem, our assumptions, and our approach and its limitations.

Motivation. Solving this subproblem of the remote localization problem is an important step toward a solution to the more general problem. Aside from the research challenge, this problem is of interest to the general public. Particular users care about their own location being determined and shared without their consent. Further, society may judge phone-based location tracking of individuals as something to be regulated or otherwise controlled. For example, in a recent report [5], the U.S. Government Accountability Office notes that federal action could help protect consumer privacy; legislatively, the proposed Location Privacy Protection Act of 2012[1] and Mobile Device Privacy Act[2] both seek to protect this information in the U.S.

Assumptions. We assume the attacker is the remote end-point of the target's communication, as is the case for streaming services such as Spotify, Pandora, and many others; or has access (perhaps unauthorized) to network-level traces at this location. We assume that the carrier, who can localize the mobile node by examining the cell towers to which it has associated, is not assisting the attacker. We assume that the attacker does not have direct access to the internals of the cellular infrastructure or to the mobile device used by the target, and therefore will find it nearly impossible [1] to geolocate the user from its carrier-assigned IP address. (Carriers use a small pool of addresses that are re-used across the country from one minute to the next.) We assume the attacker does not have the ability to direct the mobile device to reveal its location overtly. The attacker only passively analyzes the communication between the mobile device and its server. Our attacker uses only throughput measurements of the target's data stream and not the content, which could be encrypted or otherwise unavailable – though we do assume the attacker could link flows if the remote end-point IP address changes. This assumption is reasonable given our attacker model. (We enumerate limitations of our evaluation in Section 5.)

[1] `http://thomas.loc.gov/cgi-bin/query/z?c112:S.1223:`

[2] `http://markey.house.gov/document/2012/mobile-device-privacy-act-2012`

Approach. Our approach builds models of the effects of mobility upon network traffic, and uses these models to determine the mobility of users. Specifically, the attacker compares a trace of network traffic generated by a mobile user against a set of models representing specific paths through the world. The attacker creates these models by gathering information about TCP's performance on a set of possible routes that he assumes the target may take. The attacker may gather this information, which consists of traces of network traffic, during any period when the traffic observed would be similar — it need not be done strictly prior to the attack. This training information may be gathered from other users that have not disabled GPS. We conjecture that greater temporal locality will improve the attacker's performance, though we did not explicitly test this assumption.

Limitations. The traces we consider are on the order of tens of minutes long; we do not attempt to determine the shortest such trace that yields useful information. We do not attempt to solve short-duration instances of this problem, nor do we attempt to chain together small instances of the problem into a larger instance (i.e., from one instance of Wi-Fi access to a later one). We do not try to pinpoint the geographic location of the mobile, rather only the path taken from limited possibilities.

3 Data Collection and Exploratory Analysis

In this section, we describe our data and collection methodology, and we present an exploratory analysis of our data set. Our focus is on the geographic consistency of client-side signal strength and server-side throughput measurements. Our data set consists of 286 measurements of mobile phones traveling to and from a central location to four different locations, each about 25 minutes away (roughly a 360 km^2 area). We also collected 29 stationary traces, to serve as a simple baseline for our approach. In our analysis, we find that signal strength and throughput characteristics are tied to geography: in our data set, the mean throughput of 95% of 0.9 km^2 areas is statistically different from at least 90% of the other areas, and similar results hold for signal strength. The implication is that the route traveled by a mobile node is through a relatively unique sequence of mean throughputs that is classifiable.

3.1 Data Collection Methodology

Our measurements[3] are based on four Android cell phones instrumented to record traces of GPS location and signal strength. A server in our building streamed music continuously to the phones during measurement trials. We logged TCP traces at the server during trials. We later combined sets of corresponding phone and server traces, synchronizing by the timestamps within the traces. Note that it is impossible without carrier participation to take measurements at points

[3] Traces from our experiments are available for download from `http://traces.cs.umass.edu`.

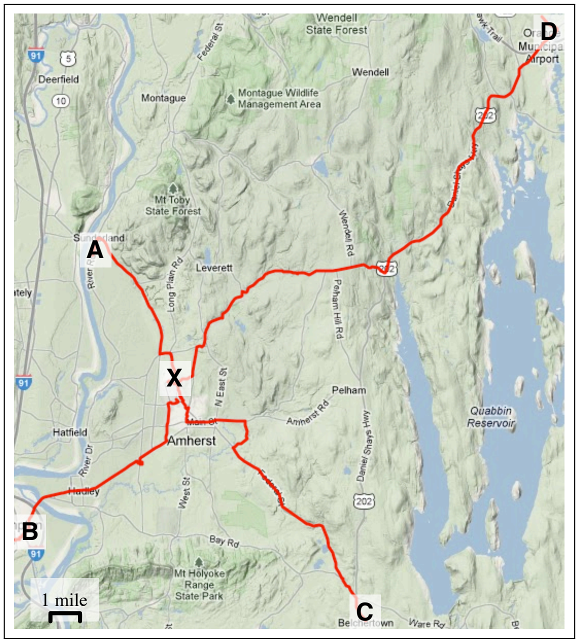


Fig. 1. We gathered data on four popular paths in our area, in two directions for each path. All paths intersect in Amherst, MA, labeled as "X".

inside the network along the path. Moreover, our goal was to take measurements that are available at the server without special access to cellular infrastructure, allowing for a weaker attacker. We took two sets of traces:

- **Mobile 3G Measurement Set**: We used *Samsung Nexus S*, *Samsung Galaxy S*, *Motorola Atrix*, and *HTC Inspire* phones, all connected to the AT&T UMTS (3G) network, to record traces. The 802.11 radio on the phone remained powered off during the experiments. We collected data during a one-month period under varying traffic and weather conditions. Each measurement was taken as a phone traveled along one of four routes going either toward or away from our central location (point X in Figure 1). The individual paths are shown on a map in Figure 1 and summary statistics appear in Figure 2. In total, we recorded 286 traces in this set.
- **Stationary 3G Measurement Set**: We recorded 29 traces from stationary phones, connected to the UMTS (3G) network, located in different locations near our central location.

The phones collected traces of GPS location (with 10m accuracy) and signal strength[4]. Each element of the traces was sampled once per second. Traces of network activity on the server consist of standard `pcap` logs. We did not limit

[4] As reported by `android.telephony.SignalStrength.getGsmSignalStrength()`.

Route	Distance (mi)	Num. Traces Collected	Throughput (KB/s) mean ± s.d.	Duration (min) mean ± s.d.
X to A	8.7	68	143.1 ± 104.3	16.4 ± 4.3
A to X	8.7	63	167.2 ± 108.2	16.1 ± 4.2
X to B	9.2	28	49.8 ± 81.8	33.6 ± 9.0
B to X	9.2	24	67.0 ± 95.8	30.5 ± 4.8
X to C	12.7	31	127.3 ± 106.8	32.1 ± 7.9
C to X	12.7	29	123.4 ± 108.5	34.8 ± 7.4
X to D	22.2	24	47.6 ± 76.2	35.0 ± 10.3
D to X	22.2	19	36.5 ± 64.3	36.1 ± 5.1
Stationary	0	29	220.4 ± 120.3	20.4 ± 5.5

Fig. 2. Details of the traces in our Measurement Sets. Letters refer to landmarks labeled in Figure 1. In total, we recorded 286 mobile and 29 stationary traces.

traces to periods of cellular connectivity, and some traces consisted of several TCP connections.

In the mobile sets, we hired several persons to collect data on these specific paths, and no person was assigned to a single path or phone. Our goal was to avoid learning the phone model or user behind the movement. Each path differed in distance, and each took about 25 minutes on average to travel by car or bus. The travel time to location A was the shortest and D the longest. We discuss the implications of path duration on classifier bias in Section 5. Because we relied on a consumer phone platform, on some occasions the experiment failed because either the end time or start time were not recorded correctly, due to a GPS failure or write-to-flash failure. We did not attempt to even out the number of traces per path after our collection period completed. Though the number of traces per route and direction varies, we did not alter which traces to collect or which to use in the experiments for any reason (most importantly, to alter classification accuracy).

3.2 Geographic Analysis

We grouped all server-side throughput and client-side signal strength measurements into small geographic areas (much small than and having no correspondence to carrier cells) to determine if each area had consistent and differentiable mean throughput. The efficacy of any throughput-based remote localization scheme depends on such consistency. We found geographic consistency in both cases and a weak correlation between the two features.

Server-side throughput is influenced by the wireless link between the phone and cell tower [3], the network conditions and infrastructure [15] between the phone and server, the TCP algorithm [10], and other factors. Signal strength is just one factor that influences the wireless link but it is the factor with the strongest tie to geography. Received signal strength is influenced by many physical

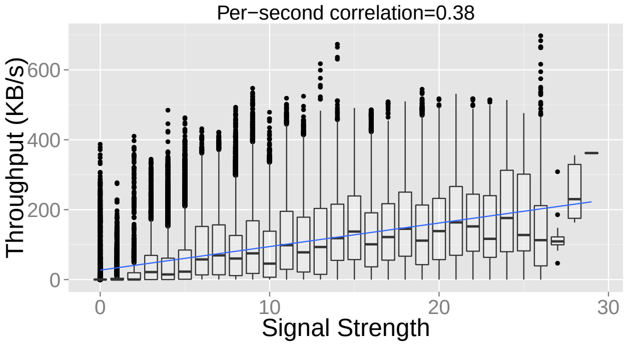


Fig. 3. On a per-second basis, the correlation between server-side throughput and client-side signal strength is 0.38. The plot shows a linear fit. Figures 13(left) and (right) show that this correlation increases to 0.58 and 0.89 on a per-trace and per-route basis, respectively.

features, including occlusions between the radio and cell tower from tree foliage, the body of the person carrying the phone, buildings, and other structures.

The range of signal strength values are integers defined in GSM standard TS 27.007, with 0 referring to -113 dBm or less, 31 referring to -51 dBm or greater, and 99 referring to an undetectable signal. Each value between 1 and 30 is a linear increase from -111 to -53 dBm. We discarded values of 31, as the range it captures is too large for a meaningful regression. We treated values of 99 as 0.

We found a weak correlation between client-side signal strength and server-side throughput of 0.38 when considered as a per-second granularity. Figure 3 shows the distribution of throughput values per signal strength value as a boxplot. The figure also plots the least-squares linear fit of the two variables as a visual guide. In Section 5, we return to this correlation on a per-trace and per-route basis, showing the correlation increases to 0.58 and 0.89, respectively.

Figure 4 shows the mean throughput (left) and signal strength (right) of geographic areas in our measurements. The error bars of each mean indicate the 95% confidence interval of the mean. Each plot is sorted by an increasing mean value, and therefore the order of areas in the plots is not the same. Using a two-sided, 95% confidence interval *t*-test, we performed a pairwise comparison of the mean throughput of the areas. 95% of areas have means that are statistically different from at least 90% of other areas. The consistency of these values and the differences among areas suggests that latent information linking throughput and geography is available for training a classifier. Signal strength measurements have a similar consistency.

In Figure 5, we plot the mean throughput of each area on a geographical grid. Mobile nodes will travel through a sequence of areas that has a unique signature of mean throughputs. The task of classification is to match the observed throughput to a training set that captures these means. In the simplest approach,

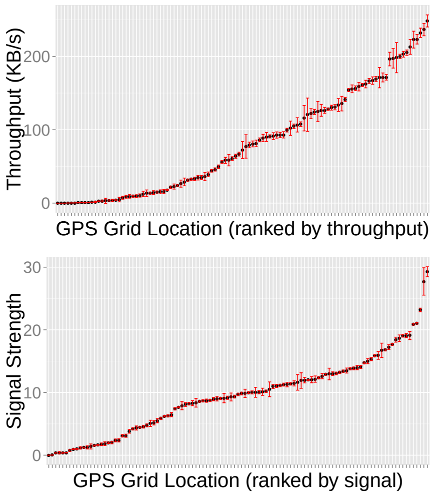


Fig. 4. The mean throughput (top) and signal strength (bottom) of geographical areas in our measurements. Each tic on the x-axis is a geographic area, and the order of areas is by increasing mean value; the order of areas in the plots is different. Error bars indicate the 95% c.i. of the mean. 95% of areas have throughput means that are statistically different from at least 90% of other areas (two-sided, 95% c.i. t-test). A similar result holds for signal strengths.

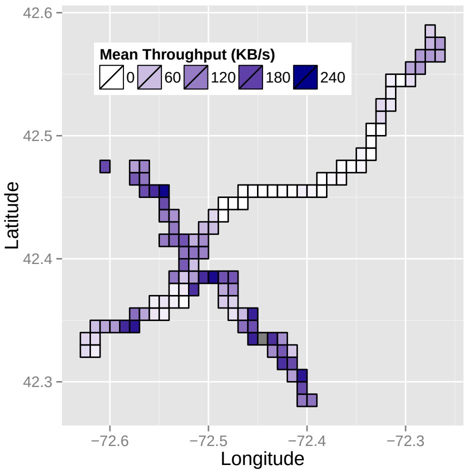


Fig. 5. The mean throughput of each area on a geographical grid. Mobile nodes will travel through a sequence of areas that has a unique signature of mean throughputs. The task of classification is to match the observed throughput to a training set that captures these means. Each area is approximately 0.9 km^2.

we can classify based on the mean throughput that a mobile device obtains from visiting a series of areas. In more advanced approaches, we can classify based on the ordered set of mean throughputs obtained, or the ordered set with timing information. We attempt all three approaches, detailed further in the next section.

4 Classifiers for Mobile Throughput Traces

Given the attacker model and the data we've described in previous sections, it remains for us to detail how an attacker can use throughput traces to determine which path a mobile phone user takes. In this section, we describe several *classification* algorithms suited to this task.

Classifiers build models of labeled training instances of data, and use these models to decide to which class an unlabeled test instance belongs. The instances we considered were created from `pcap` files, and the specific data we were interested in was TCP throughput. We discretized this data into one-second intervals, and treated each instance as a sequence X of per-chunk median throughputs

$(x_0, x_1, \ldots)$. With one second chunks, the index of each throughput value is the time since the start of the trace. The first algorithm, a straightforward k-nearest neighbor (k-NN) classifier, operates directly on these sequences. The second algorithm, based upon Hidden Markov Models (HMMs) requires some pre-processing of the sequences before operation, which we detail later.

4.1 Sequence-based k-Nearest Neighbor Classifier

We build and use a k-NN classifier as follows. We train by simply storing training instances and their labels. To classify an unlabeled instance, we first compute the instance's *distance* from each labeled training instance. Given two instances X and Y, where $\text{len}(X) \leq \text{len}(Y)$, we define

$$\text{distance}(X, Y) = \sum_{i=0}^{\text{len}(X)-1} |x_i - y_i| + \sum_{i=\text{len}(X)}^{\text{len}(Y)-1} y_i \tag{1}$$

The first summation is the per-chunk difference in throughput. The second summation is the remaining per-chunk throughputs; in effect, we are imputing zeros in the shorter trace.

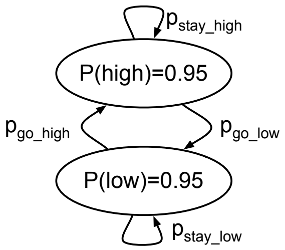
We classify the instance as the label (i.e., the route) present in the largest fraction of the k-nearest neighbors. The choice of k tunes a smoothing effect in the data: A larger k reduces erroneous labeling due to matching against outliers, while too large of a k can result in simply choosing the most common training label. We used $k = 13$ for all experiments. All values between $k = 1 \ldots 20$ performed roughly the same (59.4% to 67.1%, with a mean of 62.7%). For our data set, the choice of k is not critical. We don't expect the value is generalizable.

4.2 HMM-based Classifiers

The data consists of frequent changes in throughput, and we construct an HMM classifier that models these changes. This classifier measures the consistency and volatility of the signal at certain levels along a path.

Overview. Figure 6 shows a simple HMM, with some details elided. This HMM represents one path through the world, and the corresponding changes in throughputs that are observed along that path. In this model, there are two states, corresponding to either a high or low throughput. In each state, the corresponding symbol is emitted with probability 0.95. There is an unspecified probability, $\text{p}_{\text{stay\_...}}$, that the HMM will stay in that state, or transition to the other state with probability $\text{p}_{\text{go\_...}}$. The exact probabilities can be set manually, or trained using measured data and the Baum-Welch algorithm. In this model, a more volatile signal may show higher transition probabilities than a more consistent signal, while a consistently high signal may result in a higher $\text{p}_{\text{stay\_high}}$ and a lower $\text{p}_{\text{stay\_low}}$.

Details. We build the HMM-based classifiers as follows. We start with the *sequences* of one-second chunks of throughput corresponding to the traces. The



(a) A simple HMM before training.

Route	p stay_low	p go_high	p go_low	p stay_high
A to X	0.91	0.09	0.08	0.92
X to A	0.88	0.12	0.06	0.94
B to X	0.96	0.04	0.10	0.90
X to B	0.95	0.05	0.05	0.95

(b) HMM probabilities after training.

Fig. 6. A simple, untrained HMM as shown can be trained upon observed data. Trace data representing observed throughput is discretized into periods of high and low bandwidths, and the Baum-Welch algorithm is used to learn the most likely corresponding HMM. All emission probabilities converged to P(X)=1 from an initial setting of P(X)=0.95. For four such routes, four such two-state HMMs were learned using the Viterbi algorithm, with weights as shown in table (we used a 7-state HMM in our experiments).

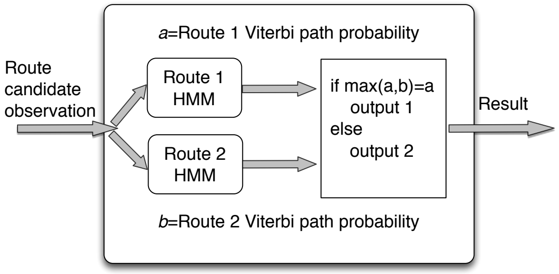

a=Route 1 Viterbi path probability
Route candidate observation
Route 1 HMM
Route 2 HMM
if max(a,b)=a
output 1
else
output 2
Result
b=Route 2 Viterbi path probability

Fig. 7. The classification approach used for labeling traces corresponding to one of two candidate paths. Per-route HMMs are trained using the Baum-Welch algorithm on labeled traces. Classification of unlabeled traces is done by choosing the route corresponding to the HMM with the highest-probability Viterbi path.

chunks are then symbolically labeled with one of n discrete symbols, where the cutoffs associated with each symbol are the equivalent quantiles. For example, $n = 2$ has a cutoff at the median and the symbols represent high and low throughputs, as illustrated in Figure 6(a). When $n = 4$, the cutoffs are at each quartile boundary, and so on. The sequence of symbols thus generated serves as input to our HMMs, as either labeled training data, or as unlabeled test data. The initial HMMs also have n states, where the state corresponding to each quantile has emission probability 0.95, and the other symbols are emitted with an equal division of the remaining 0.05 probability mass.

To train each HMM, we use the Baum-Welch algorithm, with an initial HMM as shown in Figure 6(a), where transition probabilities from each state are equiprobable. We apply Baum-Welch once for each labeled training sequence.

As an example, we show the resulting weights in Figure 6(b) for $n = 2$ to illustrate the differences between two such paths. Note that we show each direction separately, and that, as would be expected with an HMM in this scenario, the two machines for each path more closely resemble one another than those from the other path. In our traces, $n = 7$ produced best results; we tested values of n from 2 to 25. Values for n near 7 all performed nearly as well, and improvements past $n = 7$ were minimal and likely due to variance in the input data, given our relatively small data set. As with the choice of k for the k-NN classifier, our choice of n is tuned for our dataset and not generalizable.

The procedure for using our classifier is illustrated in Figure 7. We take an unlabeled sequence, and apply the Viterbi algorithm to it and each HMM, which produces an estimate of the probability that the HMM produced the observed sequence. We take the path corresponding to the most probable HMM as the prediction for that sequence.

5 Experimental Results

In this section, we report on our experiments. We begin with a brief overview of our approach and a list of our assumptions. We then describe the details of our experiments, and present and discuss our results.

5.1 Overview

Our experiments take the form of classification problems, where an attacker trains a classifier on training data, consisting of labeled sequences of throughput (training instances) as described in Section 4, and attempts to determine the class of an unlabeled instance (test instances). Varying the classifier and training and testing data allow us to determine how well the attacker can perform under different scenarios.

In all experiments, we use the standard definition of *accuracy* for a multi-class problem: the sum of correct classifications divided by the total number of classifications. Because each class is mutually exclusive as we have defined them (that is, the routes do not overlap and the classifier returns only a single result), accuracy is based on the sum of true positives for all classes and there are no true negatives possible. We trained and tested all classifiers in the same proportions on each data set, and we used leave-one-out cross-validation to measure accuracy.

Classifiers. We evaluate an attacker's accuracy using the k-nearest neighbor and throughput-based HMM classifiers described in Section 4. We also evaluate two naive approaches to classification. In the first naive approach, called *Throughput*, the classifier models each path as the mean of the mean of the throughputs associated with each path. This is among the simplest approaches to modeling a path that actually uses observed throughputs. In the other naive approach, called *Frequency*, the classifier simply chooses the class that was most common in the training data. Performing no better than Frequency, which uses no information from the data stream, implies a classifier models the situation poorly.

Summary of Experiments and Results. First, we show that an attacker can differentiate a mobile user from a stationary user, that is, make the binary choice of mobile or stationary. The attack succeeds with very high accuracy (77.4–91.2% for k-NN depending upon the exact scenario). Next, we show that given a user's starting or ending location and choice of four paths, an attacker can determine which path a user traveled, that is, choose correctly from among one of four options. This attack also succeeds with high accuracy (78.1–78.5% for k-NN depending on the scenario).

We then use our data to explore how well our method will scale to more choices. We show that given just the choice of four paths, an attacker can determine both the path and direction traveled (from among the eight possibilities) with good accuracy (63.3%). This problem parallels the problem of choosing from one of eight known paths given a starting or ending point, but is no easier: when forced to determine both path and direction, the attacker is choosing among paths where pairs are quite similar in some respects (since they are essentially mirror images of one another). We also show that in our data, the mirroring accounts for virtually none of the loss in accuracy for the k-NN classifier; for the other classifiers, the mirroring does cause a drop in accuracy.

Assumptions and Limitations. Our experiments are make many simplifying assumptions for this initial work. We assume the mobile target is always on one of n paths. We assume the attacker knows the starting location such that it can only be one of the starting points of the n paths. We assume the attacker has access to the streaming data as an end-point of the connection but cannot retrieve GPS information from the target's phone. In practice, to acquire training data, the attacker can leverage their observations of users who do not disable GPS; their travels within a cellular network and their connections to Wi-Fi provide the service with training data.

Our study is limited to one geographic location that is a small city with few tall buildings. Other small cities may be different, and cities replete with tall buildings may have very different characteristics. We don't consider situations where the user reverses direction or goes off-path. The speed of the mobile node was dictated by local traffic, but otherwise the driver went at speeds appropriate to each road. We have no data on walking or bicycling targets.

We used Subsonic[5] to stream a constant bitrate mp3 from our server to the phone, resetting the cache before each run. However, a real target might not stream data the entire length of the path (or at all), and hence we gave the attacker an advantage. Further, we did not model the complexities of commercial streaming services, which may not stream from a single location on the network. Finally, our measurements do not include any competing traffic flows to or from the phone during trace collection, which may complicate classification in practice.

[5] `http://www.subsonic.org`

Path	Classes	k-NN	HMM	Thruput	Freq
X to A	2	**83.5%**	**87.6%**	72.2%	70.1%
A to X	2	**80.4%**	**88.0%**	65.2%	68.5%
X to B	2	**91.2%**	**89.5%**	**89.5%**	50.9%
B to X	2	**77.4%**	**90.6%**	**88.7%**	54.7%
X to C	2	**83.3%**	**80.0%**	**78.3%**	51.7%
C to X	2	**84.5%**	**86.2%**	**77.6%**	50.0%
X to D	2	**81.1%**	**84.9%**	**88.7%**	54.7%
D to X	2	**87.5%**	**93.8%**	**87.5%**	60.4%

Fig. 8. Classification accuracy for differentiating *stationary* vs. *mobile users*. Bolded entries have the highest accuracy; also bolded are entries that are not statistically different from the highest rate (one-sided, two-sample proportion test; 95% c.i.). In general, mean throughput is mostly sufficient to make this binary decision, but more sophisticated techniques do slightly better.

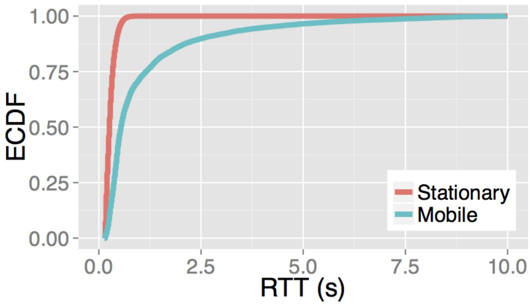


Fig. 9. The empirical CDF of server-side roundtrip time (RTT) estimates for long-running TCP connections when a mobile device is static or moving. The same device is used for both cases. The static scenario demonstrates a noticeably different distribution of estimated RTTs compared to the mobile scenario; RTT is a primary factor in TCP throughput [10].

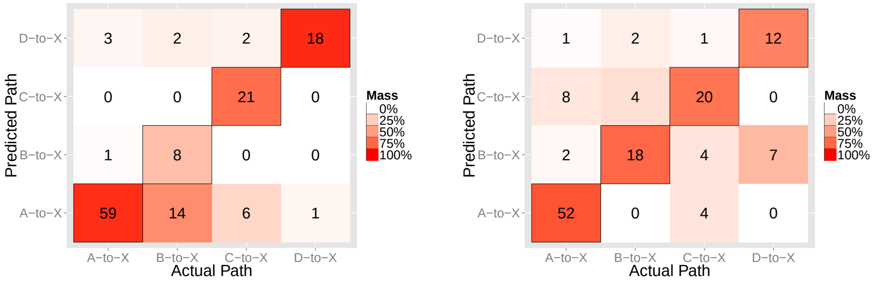
5.2 Differentiating Stationary and Mobile Users

In our first experiment, we compare our instances of data from stationary phones against those from mobile phones. In this set of experiments, all stationary traces are one class, and mobile traces corresponding to each of the routes shown in Figure 1 are treated as the other class. The results are shown in Figure 8. In that figure, we see that the Throughput classifier performs about at parity with the more sophisticated techniques.

Why does throughput-based classification work so well in this case? As we show in Figure 9, there are obvious differences in RTT estimates, and RTT is a prominent factor in TCP throughput [10]. Dramatic variations of the estimated RTT are likely the result of the increased number of local link-layer retransmissions, which seek to mitigate the impact of wireless losses on TCP [3]. These retransmissions are more common in our mobile scenarios.

Experiment	Classes	k-NN	HMM	Thruput	Frequency
4 paths × 1 direction (Inward)	4	**78.5%**	**75.6%**	67.4%	46.7%
4 paths × 1 direction (Outward)	4	**78.1%**	**70.9%**	48.3%	45.0%
4 paths × 2 directions	8	**63.3%**	39.5%	29.0%	23.8%
4 paths × either direction	4	**77.3%**	**72.7%**	57.3%	45.8%

Fig. 10. Classification accuracy for each classifier in different scenarios, discussed in Sections 5.3 and 5.4. Bold entries correspond to highest achieved values (following the same rule of significance from Figure 8).



(a) *k*-NN

(b) HMM

Fig. 11. Confusion Matrices for the *Inwards* scenarios. The number in cell (x, y) shows the count of paths of type x that were classified as path y using our (a) k-NN and (b) HMM-based classifiers. Cells are shaded to show the mass of each distribution (by column); a perfect classifier would have non-zero entries on only the diagonal.

5.3 Determining a User's Path

In our next set of experiments, we assume that user's starting or ending location is known, and that our goal is to determine which of four paths were taken by the mobile. Specifically, we train and test classifiers using only traces that are *Inward* to the central location ("…to X") described in Section 3; then we do so again, using only *Outward* ("X to …"). Each set of experiments thus considers four classes. The results of these experiments are shown in Figure 10.

In these experiments, the k-NN and HMM-based classifiers significantly outperform the naive classifiers. In line with our intuition, we find that throughput alone does not adequately differentiate routes.

Figure 11 provides finer details on the k-NN and HMM classifier results for the *Inward* paths. While the classifiers do well in labeling most paths correctly, two particular types of error stand out. The k-NN classifier misclassifies many instances into the A-to-X class. This error is partly because throughput for a path may have an increased variance at times, particularly in areas of unstable signal. Since A-to-X is the most common class, it tends to win a majority in the voting if certain parts of a trace don't clearly match another class.

The HMM tends to conflate paths involving endpoints A and C, and paths involving B and D. Upon further analysis, we realized that these two paths

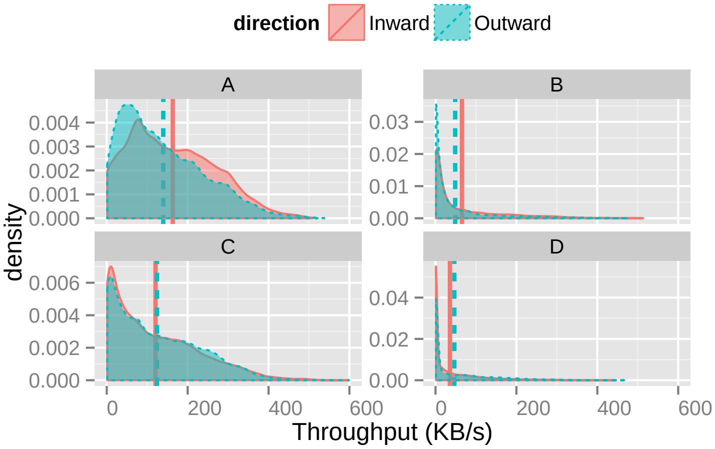


Fig. 12. Per-route distribution of throughput across all traces gathered on each path shown in Figure 1. Vertical lines denote the mean value for the corresponding distribution. Since per-route differences in throughput are the basis for classification, routes with similar distributions are more likely to be misclassified as one another, such as routes involving B and D in our experiments.

contained instances that include long periods of high throughput (in the former case) or little to no throughput (in the latter) and thus there is more similarity between the HMMs trained for those classes. This effect can be seen in the distribution of throughputs for these routes in Figure 12. In general, paths that cover the same geographical area in different directions exhibit similar throughput distributions, resulting in a lowering of accuracy for classifiers that do not consider sequence information.

As we discussed in Section 3, the reason classification is possible is that throughput is geographically consistent in our dataset (See Figure 5). The experiments in this section demonstrate that path-level classification can make meaningful use of such consistency. Further, we see a correlation of signal strength and throughput at a path- and trace-level of granularity that is stronger than at the per-second granularity reported in Figure 3. The mean throughput observed at the server for each route has strong positive correlation of 0.89 with the median signal strength on that route. Figure 13(left) shows the linear fit for each of the eight paths in our mobile traces. The high correlation between per-route mean throughput and median signal strength suggests that route classification based on throughput means or distributions — as performed using the k-NN, HMM, or Closest Mean Throughput method — should capture some information about the location of the mobile phone.

We found a positive correlation of 0.58 at a per-trace level, shown in Figure 13(right). This drop in correlation — and the even smaller correlation of

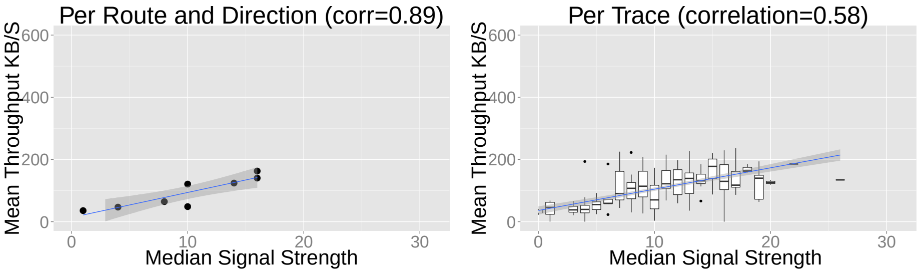


Fig. 13. The mean throughput for each route has strong positive correlation with the median signal strength on that route. The correlation coefficient is 0.89 per-route-and-direction, and 0.58 per-trace. (Note that it was 0.38 in Figure 3, which was a per-second granularity.) The correlation increases as granularity decreases, which supports our hypothesis that there are consistent effects of geography upon throughput.

Path	Classes	k-NN	HMM	Thruput	Frequency
A	2	**74.8%**	64.1%	60.3%	51.9%
B	2	**84.6%**	67.3%	57.7%	53.8%
C	2	**76.7%**	31.7%	46.7%	51.7%
D	2	**79.1%**	51.2%	**65.1%**	55.8%

Fig. 14. Classifier accuracy when determining direction of travel on a given path. Bold entries correspond to highest achieved categories (following the same rule of significance from Figure 8). Only the k-NN classifier includes sequence information in its model of paths, and it is consistently in the highest-achieving category.

0.38 at the per-second level — speaks to the challenge of this task: network performance is generally consistent for a path but it weakens significantly for shorter time scales. Additional features from the network traffic are likely needed to advance classification accuracy to work with finer time scales or geographies. In the next subsection, we investigate how these classifiers scale.

5.4 Scaling and Direction-Insensitivity

Do our techniques scale beyond choosing one of four paths? While we cannot answer this question generally, we can artificially increase the number of choices the attacker has by considering each path in each direction as a separate class, giving us eight possible classes corresponding to the eight routes in Figure 2.

Our results are shown in the “2 directions” row of Figure 10. The accuracy of all methods except for the k-NN classifier collapses: it achieves a reasonable 63.3%, statistically significantly above the others. The HMM classifier at 39.5% scores statistically significantly above the Throughput and Frequency classifiers at 29.0% and 23.8%; the latter two are not significantly different from one another. Two questions arise: Why does performance decrease with more options? And why does the k-NN classifier do so much better than the others?

A key way that the k-NN classifier differs from the other three is that it considers the sequence information, whereas the other three discard or condense it. We conjecture that this additional information enables the k-NN classifier to perform well when some paths are essentially mirror images of one another, and we explore this conjecture with two more experiments.

First, we trained and tested the classifiers on each of the four paths in a direction-insensitive manner; that is, we gave paths from A to X and from X to A the same label (simply "A"), and so on. The results, in the row labeled "either direction" of Figure 10, show that the classifiers return to their previous level of accuracy in this scenario. Second, we trained and tested each classifier on four new scenarios. Each was a binary classification test where the classes were either paths from A to X or from X to A; then either from B to X or from X to B as a separate experiment; and so on. The results are shown in Figure 14; we see that the k-NN classifier consistently performs well, and the other classifiers are not able to distinguish the direction of a path.

Finally, a note on classifier bias. Because we could not control the duration of the paths, we verified that our sequence-based (k-NN) classifier did not perform well merely because paths were different lengths. We implemented a length-based classifier for this purpose (i.e., one that classified based on the duration of a trace). We found that while the length-based classifier performed better than frequency, the accuracy was driven by a different set of paths than the k-NN's correctly classified paths.

5.5 Approaches to Enhancing Privacy

We did not test any approaches for enhancing the privacy of users against this attack, but many existing techniques are likely to be effective. To prevent revealing their travel paths to nosy remote servers, phone users will need to traffic shape or otherwise perturb their data transmission, incurring a performance penalty.

For example, a trusted proxy located outside the cellular network can re-shape traffic before reaching the attacker. As noted in Section 2, we assume the carrier is not assisting the attacker. The proxy could be set up as a VPN, which we suggest not for the encryption but because it is a protocol widely supported by smart phones as a transparent method of redirecting traffic. It is feasible that the mobile device could reshape the traffic on its own. Most simply, it could limit throughput to a peak level that is reasonable across a wide area of the cell network. Or it could enforce regions of zero throughput. Of course, the challenge is to shape traffic in a way that does not overall reduce throughput or the interactivity needed by the application. This type of shaping may be easy for bulk file transfer (low interactivity required), moderately challenging for web browsing (where caching and pre-fetching may help mask throughput ceilings), and very challenging for interactive audio and video calls (where users are most sensitive to throughput limitations and network delay jitter).

6 Related Work

Mobile Phone Localization. Precisely localizing mobile phones or other similar devices on the basis of GSM and other location-explicit information is an active area of research, however, these works use information available only to the mobile user (such as which 802.11 base stations or cell towers are in range [6,11]) or their carrier (such as the pattern of handoffs [2] or other administrative details [14]). In our work, we focus on *remotely* localizing another party based only on a TCP traffic stream rather than local information.

Kune et al. [8] propose a technique to test if a user is present within a small area or absent from a large area by simply listening on the broadcast GSM channel. The focus of their work is on lower layers of GSM communication stack. We did not extend our study to analyze lower layers of 3G, because it is a legal violation in our jurisdiction. And again, our study is concerned with remote observation of network streams over cellular links, and largely treats the cellular infrastructure as a black box.

Xu et al. [14] present an approach for localizing performance measurements in 3G networks. They exploit the predictability of users' mobility pattern to develop a clustering algorithm for grouping related cell sectors and assigning IP performance measurements to fine-grained geographic regions. The proposed technique requires access to the cellular infrastructure. In contrast, our technique for network-based localization requires remote passive observation of the target, and data collection independent from the target and internals of the infrastructure.

Balakrishnan et al. [1] show that individual cell phones can expose different IP addresses to servers within time spans of a few minutes, and find that IP-based geo-localization is "impossible" in cellular networks. They show that application-level latencies can differ greatly among cities thousands of miles apart. Moreover, they show that the variation of latencies in short time spans is not high. Our work is complementary and extends similar notions further: we show the consistency of throughput at the finer granularity of square-kilometer regions, and we demonstrate successful classification experiments using such features.

Xu et al. [15] show that in contrast to wired Internet traffic, current cellular data traffic traverses through a limited number (4–6) of Gateway GPRS Support Node (GGSNs), which is the first IP hop of a data connection. The authors show that local DNS servers provide an appropriate approximation to estimate a user's network location (i.e., one of the 6 GGNs) for purposes of mobile content placement and server selection, due to the restricted routing in cellular networks. By assuming availability of partial information about the possible routes a user could be on, our approach aims at a much more granular localization than the DNS method proposed by Xu et al., which is limited to finding approximate network locations.

Other Remote Attacks. Kohno et al. [7] present a technique for fingerprinting a physical device remotely by exploiting clock skews. Their approach could be used to remotely identify the same device connected to the Internet at different times or using different IP addresses. Our approach, which is focused on detecting the routes taken by a mobile node, is orthogonal to this work and could benefit from it when locating the end-points of a target's travel path.

NAT and firewall policies of cellular carriers are explored in the work by Wang et al. [12]. They identify a set of such policies that directly impact performance, energy, and security of mobile devices. For instance, they show that NAT boxes and firewalls set timeouts for idle TCP connections, which sometimes lead to significant waste of energy on the mobile device. The authors show that in spite of deployment of firewalls, cellular networks are still vulnerable to denial of service and battery draining attacks. We explore another type of attack on the location privacy of mobile cellular users.

Case Law. Can the U.S. government, as a third party, obtain the throughput information needed for remote localization? Recently, in *U.S. v. Jones (No. 101259)*, the Supreme Court ruled that law enforcement need a warrant to remotely track a person's location through prolonged use of a GPS device. In that case, a GPS device was placed by the government on the target's car, which was considered a trespass by the majority opinion. In contrast, in our scenario the government would obtain only the size of packets received by a third party, assuming the government already knew through physical observation the starting or ending point of a target's journey. Acquisition of such signaling information does not usually require a warrant or wiretap because it is not content (see *Smith v. Maryland, 442 U.S. 735 (1979)*). Accordingly, recent cases involving acquisition of cell site information may be more relevant than *Jones*, but the issue is unsettled. For a summary of recent rulings on cell sites, see footnotes 8 and 9 in an opinion[6] from the *Jones* retrial.

7 Conclusion and Open Problems

We have demonstrated that the patterns of data transmission between a server on the Internet and a moving cell phone can reveal the geographic travel path of that phone. While the GPS and location-awareness features on phones explicitly share this information, phone users will likely be surprised to learn that disabling these features does not suffice to prevent a remote server from determining their general mobility. Our work shows that a simple k-nearest neighbor classifier can discover and exploit features of the geography surrounding possible travel paths to determine the path a phone took, using only data visible at the remote server on the Internet and training data collected independently.

It is an open and important problem to quantify the extent to which a user's location can be compromised in this fashion — with greater accuracy and among larger numbers of paths and different geographies — and to determine just how much information is needed to make these inferences. We conjecture that preprocessing of the data — smoothing or aligning the sequences — may more

[6] `http://www.gpo.gov/fdsys/pkg/USCOURTS-dcd-1_05-cr-00386/pdf/USCOURTS-dcd-1_05-cr-00386-9.pdf`

readily reveal patterns within a given path. Other algorithms may also improve the attacker's performance: for example, we have implemented a discrete left-right HMM, which models the sequence of changing throughput distributions along a path, with limited success. We anticipate that performance may improve with smoothed data or a more apt model, such as one with continuous observations. A model representing a user's position in 2-D space rather than along a specific path would allow an attacker to more easily identify user location in a real-world scenario. Finally, the ability to classify shorter traces would allow an attacker to deduce location with finer granularity.

Acknowledgements. This work was supported in part by NSF award CNS-0905349. We thank Mark Corner for early discussions of this work.

References

1. Balakrishnan, M., Mohomed, I., Ramasubramanian, V.: Where's that phone? Geolocating IP addresses on 3G networks. In: ACM IMC. pp. 294–300 (2009)
2. Becker, R.A., Caceres, R., Hanson, K., Loh, J.M., Urbanek, S., Varshavsky, A., Volinsky, C.: Route classification using cellular handoff patterns. In: ACM UbiComp. pp. 123–132 (2011)
3. Chan, M.C., Ramjee, R.: TCP/IP performance over 3G wireless links with rate and delay variation. In: ACM MobiCom. pp. 71–82 (2002)
4. Franken, A.: The Location Privacy Protection Act of 2011 (S. 1223) information sheet. `http://www.franken.senate.gov/files/documents/121011_LocationPrivacyProtection.pdf` (2011)
5. Goldstein, M.L., et al.: Mobile Device Location Data (United States Government Accountability Office). `http://www.gao.gov/assets/650/648044.pdf` (Sept 2012)
6. Hightower, J., LaMarca, A., Smith, I.: Practical Lessons from Place Lab. IEEE Pervasive Computing 5(3), 32 –39 (July-Sept 2006)
7. Kohno, T., Broido, A., Claffy, K.: Remote physical device fingerprinting. IEEE Trans. on Dependable and Secure Computing 2(2), 93–108 (May 2005)
8. Kune, D.F., Koelndorfer, J., Hopper, N., Kim, Y.: Location leaks on the GSM Air Interface. In: ISOC NDSS (Feb 2012)
9. Open Standards for Real-Time Bidding (RTB): OpenRTB Mobile RTB API v1.0. `https://code.google.com/p/openrtb/downloads` (Feb 2011)
10. Padhye, J., Firoiu, V., Towsley, D.F., Kurose, J.F.: Modeling TCP Reno Performance. IEEE/ACM Trans. Netw. 8(2), 133–145 (Apr 2000)
11. Thiagarajan, A., Ravindranath, L., Balakrishnan, H., Madden, S., Girod, L.: Accurate, Low-Energy Trajectory Mapping for Mobile Devices. In: USENIX NSDI (Mar 2011)
12. Wang, Z., Qian, Z., Xu, Q., Mao, Z., Zhang, M.: An untold story of middleboxes in cellular networks. In: ACM SIGCOMM. pp. 374–385 (Aug 2011)
13. Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. SIGKDD Explor. Newsl. 12(1), 40–48 (Nov 2010)
14. Xu, Q., Gerber, A., Mao, Z., Pang, J.: AccuLoc: practical localization of performance measurements in 3G networks. In: ACM MobiSys. pp. 183–196 (Aug 2011)
15. Xu, Q., Huang, J., Wang, Z., Qian, F., Gerber, A., Mao, Z.: Cellular data network infrastructure characterization and implication on mobile content placement. In: ACM SIGMETRICS. pp. 317–328 (2011)