Tracking Vehicular Speed Variations by Warping Mobile Phone Signal Strengths

Gayathri Chandrasekaran*, Tam Vu* Alexander Varshavsky†, Marco Gruteser*,
Richard P. Martin*, Jie Yang‡, Yingying Chen‡

* WINLAB, Rutgers University
North Brunswick, NJ 08902
{*chandrga, tamvu, gruteser, rmartin*}
@winlab.rutgers.edu

† AT&T Labs
Florham Park, NJ 07932
{*varshavsky*}
@research.att.com

‡ Stevens Institute of Technology
Hoboken, NJ 07030
{*jyang, yingying.chen*}
@stevens.edu

***Abstract*—In this paper, we consider the problem of tracking fine-grained speeds variations of vehicles using signal strength traces from GSM enabled phones. Existing speed estimation techniques using mobile phone signals can provide longer-term speed averages but cannot track short-term speed variations. Understanding short-term speed variations, however, is important in a variety of traffic engineering applications—for example, it may help distinguish slow speeds due to traffic lights from traffic congestion when collecting real time traffic information. Using mobile phones in such applications is particularly attractive because it can be readily obtained from a large number of vehicles.**

Our approach is founded on the observation that the large-scale path loss and shadow fading components of signal strength readings (signal profile) obtained from the mobile phone on any given road segment appears similar over multiple trips along the same road segment except for distortions along the time axis due to speed variations. We therefore propose a speed tracking technique that uses a Derivative Dynamic Time Warping (DDTW) algorithm to realign a given signal profile with a known training profile from the same road. The speed tracking technique than translates the warping path (i.e., the degree of stretching and compressing needed for alignment) into an estimated speed trace. Using 6.4 hours of GSM signal strength traces collected from a vehicle, we show that our algorithm can estimate vehicular speeds with a median error of $\pm\ 4mph$ compared to that of using a GPS and can capture significant speed variations on road segments with a precision of 68% and a recall of 84%.

***Keywords*-Derivative Dynamic Time Warping (DDTW), Received Signal Strength (RSS), Global System for Mobile Communications (GSM)**

I. Introduction

This paper considers the problem of estimating fine-grained speed and detecting temporary speed variations of a vehicle from cellular handset signals. More fine-grained speed traces could benefit a number of transportation applications. For example, fine-grained speed trace could improve estimating and pinpointing traffic congestion, particularly on arterial roads with traffic signals. Since fine-grained speed traces reveal where on a road segment vehicles slow down, it becomes easier to distinguish speed variations due to congestion from slowdowns due to red traffic lights. Fine-grained speed traces also reveal whether traffic is flowing slow but smoothly or in a stop-and-go fashion. It can also show where frequent lane changes occur that cause traffic shock waves. These factors have a significant effect on accident rates and gasoline consumption, and would therefore be important to monitor on a larger scale. Techniques to determine vehicle speed from cell phone signals are particularly useful because they do not incur the high infrastructure costs of traffic cameras or loop detectors embedded into the roadway [7], [6], [11]. While fine-grained speed traces can also be obtained through networked in-vehicle GPS devices, cell phone signals can readily be collected from a much larger number of vehicles. Cell phone signal strength readings also impose no energy overhead, at least when collected at the base station. Existing speed estimation techniques from cell phone communications are limited to estimating average speeds over road segments. One approach derives speed from the time between two handoffs [21], [10]. In our own prior work [4], we have also shown how average speeds can be estimated by matching a cell phone's signal strength trace against a known trace from this road segment. While these solutions can cover most of the arterial roads, the average speed estimates are typically over road segments of about 100m and cannot track vehicle's exact speed variations.
Our Approach. In this paper, we propose a speed trace estimation technique based on a Derivative Dynamic Time Warping (DDTW) algorithm that aligns a received signal strength (RSS) trace from a moving cell phone handset with a reference trace for a given road segment to estimate speed. The technique relies on the observation that large scale path loss and dominant shadow fading effects usually remain quite constant at the same location. To illustrate this insight, Fig. 1 plots the instantaneous speed and RSS trace from the associate cell tower for two vehicle trips along the same stretch of a road. The vehicle drove roughly at the same speed during the first 150 seconds of both trips, but then it slowed down in the first trip and sped up in the second.[1] The graph shows how the RSS traces remain similar over the first part of the trace,

[1]The car travelled the same distance in both cases and stopped at the same physical location. However, due to the speed difference, the first trip took about 300 seconds while the second only lasted 200 seconds.

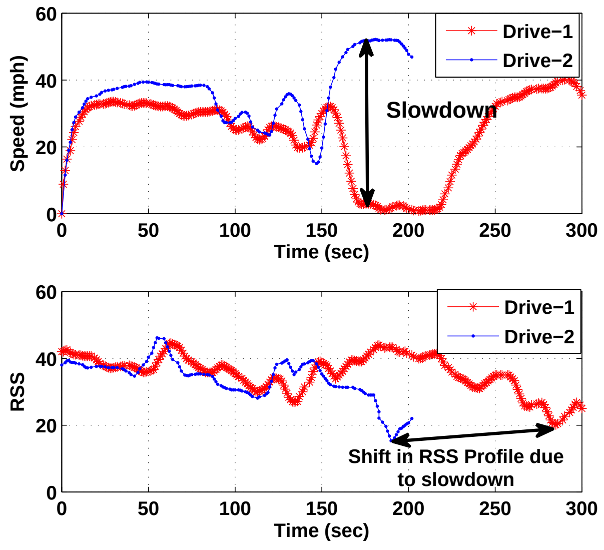


Fig. 1. Stability of RSS over time

where the vehicle traveled at the same speed, and depart when the vehicle varied its speed in the later part of these two trips. Note also, how the trace from the slower trip is essentially a stretched version of the faster trip in the second part of the trace. For example, the dip below an RSS value of 20 dB, occurs in the same location in both trips but due to the speed difference the graph shows them at different times.

The key idea underlying our technique is to stretch (or compress) the RSS trace until it best matches the reference trace. The stretch factor can vary over the length of the trace. Since the instantaneous speed over the reference trace is known, the algorithm can then convert these stretch factors into instantaneous speed estimates for the test RSS trace. Thus, our approach can perform vehicular speed tracking and detect bottlenecks on road segments effectively. We assume that training RSS profiles and their speeds are available for road segments under study. These could be collected as part of the service provider signal measurements to determine coverage. We also assume that the approximate starting location and the road segment the vehicle travels on is known, for example by monitoring handoff locations as shown in prior work [12].

The rest of the paper is organized as follows: We first perform an overview of general vehicular speed estimation techniques in Section II We then describe our technical approach in terms of DDTW algorithm, and explain how speed estimation and tracking is performed in Section III. We next present how to predict road bottlenecks by detecting vehicular slowdowns in Section IV. We collect traces from real traffic drives and compare the performance of our mechanism with existing algorthms in Section V, then conclude in Section VI.

II. Background

In this section, we first review existing studies on vehicular speed estimation. We then describe the baseline algorithms that we compare with our approach in this work.

The existing work on vehicular speed estimation can be classified based on the modality of sensing: *Fixed Infrastructure based sensing*, *Smartphone based sensing*, *Cellular phone based sensing*, and *Doppler shift-based sensing*.

Fixed Infrastructure based sensing: By far the most common of highway speed estimation system is the inductive loop detectors [7], [6], [11] which are based on on-road sensors embedded in the pavement. Traffic cameras [9] have also been installed on roads that uses a sequence of image captured on several cameras on the road to calibrate the speed of a moving vehicle. [7] has shown that speed estimation accuracy using loop detectors for a vehicle traveling at over 50mph can be in the order of 20mph to 120mph. Besides that, they suffer from their limited reliability and high installation cost, which makes it hard to maintain significant coverage on the road network.

Smartphone based sensing: Using GPS enabled smartphones for sensing [12], [1] has gained huge popularity in the recent times due to its negligible deployment cost. These techniques, if adopted by a large number of users, can provide very accurate speed estimation on most roadways. However, frequent sampling of the GPS unit can result in fast battery drain on the mobile phone. [22] tried to overcome some of the energy limitations by sub-sampling the GPS and combining the Wi-Fi outdoor positioning along with map-matching to estimate speeds with high accuracy. Still, energy consumption remains higher than approaches that use existing phone signals. It also require software modifications on each handset which makes bootstrapping the service more difficult.

Cellular phone based sensing: Unlike the smartphone based sensing, these techniques rely on the location of the cellular phone over time calibrated either using triangulation of the GSM signal strength [24] over time or Fingerprint matching of the phone successive signal strength readings [5] or the location where the cellular phone handsoff between towers [21], [10]. Our previous work [4] uses the stability in signal strenth profiles on a road segment to derive average speed estimate. [20] uses the rate of change of RSS between successive samples to determine the speed. While all of the above techniques can overcome the bootstrapping (since the provider already has access to the signal strength information from phone) and energy issues that were present in smartphone based sension, these can only estimate average speeds over segment of length typically over 100m. Therefore, they cannot be used to track small variations in speed that are important for several traffic engineering applications. We differ from all the above techniques by estimating speeds with high accuracy. In addition, we are the first to show the possibility of using GSM signal strength for tracking temporary speed variations (for, example bottlenecks causing slowdowns).

Doppler shift-based sensing: Finally, [23], [25] makes use of the doppler shift in frequency caused by the moving transmitter to estimate speed. [23] can only perform coarse speed classification while [25] can predict the actual speed of the mobile. But the latter assumes the presence of strong Line of Sight(LOS) component between the transmitter and the receiver which can make this technique impractical.

We further choose two representative algorithms, *Localization Algorithm* and *Normalized Euclidean Distance Algorithm*, which can be used to perform vehicular speed tracking and detect the bottlenecks in road segments. These two algorithms will be used as the baseline approaches to compare with our mechanism in Section V. Note that the performance of the localization algorithms for tracking speed variations are similar to our prior speed estimation algorithm [4]. We therefore only include the more general and better known localization algorithm as a baseline algorithm.

Localization Algorithm: This method estimates the speed of a mobile phone between two points by estimating the phones' locations at the two points, calculating the distance the phone has travelled and dividing it by the time travelled. In this paper, we use the fingerprinting [17] algorithm for determining phone's location. The algorithm uses the RSS fingerprints obtained from 7 neighboring towers at different known locations as the training. When an RSS fingerprint is obtained from a mobile at an unknown location, the algorithm estimates the euclidean distance in signal space between this obtained fingerprint and all the training fingerprints and determines the location to be the location of the training fingerprint that yields the minimum euclidean distance.

Normalized Euclidean Distance Algorithm: This algorithm detects speed changes during speed tracking, e.g., slowdowns, by calculating the normalized euclidean distance between consecutive GSM measurements and declaring a slowdown when the distance falls beyond a certain threshold.The normalized Euclidean distance between two measurements A and B, having n common cell towers is defined as:

$$\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}/n \quad (1)$$

Note that Euclidean distance between successive samples from a mobile phone is directly proportional to the distance the phone moves in physical space, which in turn depends on how fast the phone moves. While we cannot derive an accurate speed estimate from this relation, we can still predict regions where there are slowdowns. We experimented with multiple other metrics suggested in [20], but found the normalized euclidean distance to work the best. Hence, we chose to use this algorithm for comparison with our mechanism.

III. Speed Tracking

Our speed tracking technique comprises two components. First, the Derivative Dynamic Time Warping Algorithm (DDTW) algorithm aligns a given signal profile with a known training profile. Second, the speed tracker will convert a warping path produced by the DDTW algorithm into a speed trace for the vehicle. This approach assumes that a training signal profile from the same road segment is available, which was collected at a known speed. Similar measurements are often carried out by cellular service providers to create cellular coverage maps. The testing signal profiles can be either obtained on a cell phone handset itself, or at the base station (if the cell phone is on an active call). The speed trace produced by this technique can then also serve as input for the slowdown estimation technique described in the following section.

A. Derivative Dynamic Time Warping Algorithm

To find the optimal alignment between sequences of signal strength measurements for speed estimation, we apply two sequences of signal strength measurements, one called the training and the other called the testing, to the Dynamic Time Warping (DTW) Algorithm. Dynamic Time Warping is a classic dynamic programming algorithm which has been widely used for optimal alignment of two time series datasets and was particularly popular for applications like speech processing[15], [19], data mining[16], [14], and gesture recognition[8].

In particular, we use a variant of the DTW algorithm called Derivative Dynamic Time Warping (DDTW) [13], which exploits the same principle as DTW but for the input data, where, instead of the time-series of RSS, we use the time-series of derivative of RSS. For example, if $A = (a_1, a_2, ...a_M)$ is a time series of RSS measurements collected over M time points, the input to DDTW is $A' = (a'_1, a'_2, ...a'_M)$, the derivative of A which is defined as

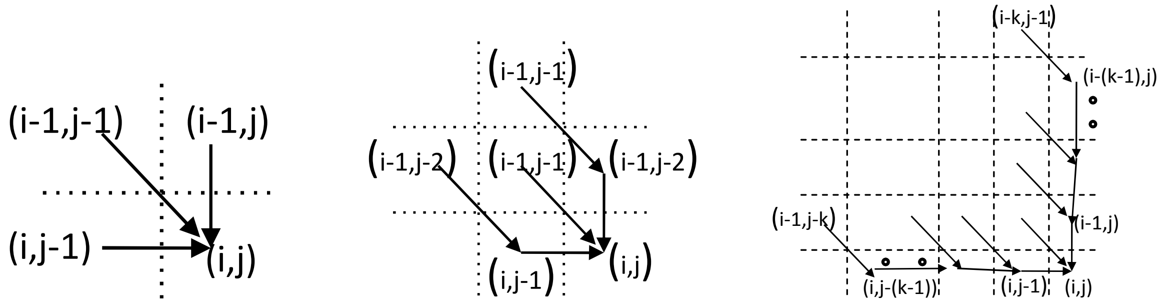
$$a'_i = \frac{(a_i - a_{i-1}) + (a_{i+1} - a_{i-1})/2}{2} \quad 1 < i < M. \quad (2)$$

Given two RSS profiles - A and B with lengths of M and N samples respectively, DDTW constructs a distance matrix $d[M \times N]$ which is defined as:

$$d(i,j) = (a'_i - b'_j)^2 \quad (3)$$

where a'_i and b'_j are the i^{th} and j^{th} elements of the derivative of the RSS profiles A and B respectively. With this $d[M \times N]$ as the input to the algorithm, DDTW returns a warping path $P = (p_1, p_2, ...p_k)$ where $p_l = (x, y) \in [1 : M] \times [1 : N]$ for $l \in [1 : k]$ as shown in Figure 3. The warping path must satisfy the following conditions:

1) **Boundary Condition**: $p_1 = (1, 1)$ and $p_k = (M, N)$. This ensures that the warping path always starts at $(1, 1)$ and ends at (M, N).
2) **Monotonicity Condition**: If $p_{k-1} = (c, d)$ and $p_k = (e, f)$, we have $e - c \geq 0$ and $f - d \geq 0$. The monotonicity condition ensures that the matching always progresses in the forward direction of time.
3) **Global Constraints**: Global constraints are constraints that limit the region in which the warping path can exist. In addtion, global path constraints also guarantee the existance of a path from $(1, 1)$ to (M, N). Figure 3 illustrates the region for warping path generation. The region enclosed within the parallelogram is the region that corresponds to the global constraints. In Figure 3, E_{MAX} is defined as the maximum allowable expansion (or compression) in time axes of one time series with respect to the other, and is chosen to be $max(2, \lceil \max(M, N)/\min(M, N) \rceil)$. The ratio $\lceil \max(M, N)/\min(M, N) \rceil$ defines the amount of expansion of one trace relative to the other. Accordingly, the sides of the parallelogram are set to have slope values of E_{MAX} and $1/E_{MAX}$.
4) **Local Constraints**: Finally, Local constraints define the set of admissible step-patterns. There are three types of



(a) Unconstrained (b) Constrained, $E_{MAX} = 2$ (c) Constrained, $E_{MAX} = k$

Fig. 2. DDTW local constraints that restrict the admissible paths to every location within the matrix:
(a) $C(i, j) = d(i, j) + \min(C(i-1, j-1), C(i, j-1), C(i-1, j))$
(b) $C(i, j) = d(i, j) + \min(C(i-1, j-1), C(i-1, j-2) + d(i, j-1), C(i-2, j-1) + d(i-1, j))$
(c) $C(i, j) = \min\Big[\min_{1\le r\le E_{MAX}}\Big(C(i-1, j-r) + \sum_{j_1=j-(r-1)}^{j} d(i, j_1)\Big), \min_{2\le r\le E_{MAX}}\Big(C(i-r, j-1) + \sum_{i_1=i-(r-1)}^{i} d(i_1, j)\Big)\Big]$

step progression: *horizontal*, *vertical* and *diagonal*. As shown in Figure 2, different kinds of local constraints are possible. For example, Figure 2(a) shows the most unrestrictive step constraint where (i, j) can be reached from one of its three neighbours $(i-1, j-1), (i-1, j), (i, j-1)$. Whereas Figure 2(b) and 2(c) illustrate more constrained progressions where a diagonal progression is forced for every E_{MAX} horizontal or vertical progressions.

To generate a warping path, DDTW constructs a cost matrix $C[M \times N]$ which represents the minimum cost to reach any point (i, j) in the matrix from $(1, 1)$ using a dynamic programming formulation. For example, in Figure 2(a), (i, j) can be reached from one of its three neighbours, namely, $(i-1, j-1)$,$(i-1, j)$, and $(i, j-1)$, and the algorithm picks the neighbour that has the minimum cost. This relation can be shown as:

$$C(i, j) = d(i, j) + \min(C(i-1, j-1), C(i, j-1), C(i-1, j)). \tag{4}$$

However, using an unconstrained local constraint as shown in 2(a) can lead to an undesirable effect called "singularities" [13] where either one sample point in the testing is mapped to a very large number of samples in training (unrestricted horizontal progression) or many points in testing map to the same point in training (unrestricted veritical progression). This effect as observed previously[15] can be minimized by using a more constrained topology for forward progression. In this work, we thus take an approach of using the constrained DDTW with a maximum expansion of E_{MAX}. our local constraints resemble the ones in Figure 2(b) and 2(c). For example, 2(b) forces a diagonal progression before every horizontal or vetical progression, whereas 2(c) allows upto (E_{MAX}) horizontal or vetical progressions before forcing a diagonal progression. For a complete description of local constraints, we refer the readers to [15]. The local constraints that we use in this work allow upto E_{MAX} vertical or horizontal progressions before forcing a diagonal progression and the cost matrix $C(i, j)$ corresponding to this local constraint can be formulated as

$$C(i,j) = \min\Big[\min_{1\le r\le E_{MAX}}\Big(C(i-1,j-r) + \sum_{j_1=j-(r-1)}^{j} d(i,j_1)\Big), \min_{2\le r\le E_{MAX}}\Big(C(i-r,j-1) + \sum_{i_1=i-(r-1)}^{i} C(i_1,j)\Big)\Big]. \tag{5}$$

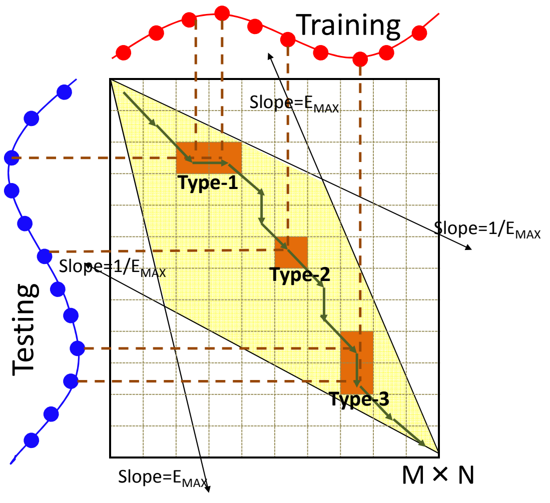
Note that the optimal path to (i, j) depends only on the values of (i', j') where $i' \le i$ and $j' \le j$. From the cost matrix, the algorithm derives a warping path P by back-tracking the constructed cost matrix from (M, N) to $(1, 1)$. While backtracking, the path that the algorithm chooses from any point (i, j) will be the (i', j') that resulted in optimal $C(i, j)$. We will next explain how we use the warping path to estimate the speed of the testing trace.

B. Estimating Vehicular Speed from DDTW's warping path

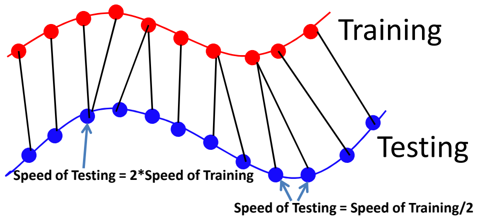
The DDTW algorithm returns a warping path P between the points $(1, 1)$ to (M, N). This warping path defines the optimal alignment between the two time series, which in this case are the RSS measurements from the training and testing drives. As explained in Section I, there is a direct correlation between the speed of vehicle and the overall shape of the RSS curve. Therefore, an optimal alignment of the RSS curves from the training and testing drives can yield a corresponding speed estimate of one drive relative to the other.

We define three kinds of matching between training and testing traces: *Type-1*, *Type-2*, and *Type-3*. If one point in the testing trace is mapped to k points in the training trace as shown in Figure 3(b), the resulting speed estimate for the testing trace is k times that of the training. We call this as Type-1 matching as shown in Figure 3(a). The figures illustrate this for $k = 2$. Similarly, Type-3 matching is when k points in the testing trace are mapped to one point in the training trace, speed of the testing trace is $1/k$ times the training speed. Finally, Type-2 match is when one point in testing maps to exactly one point in training trace. In this case, speed of testing equals speed of training.

We note that the estimated speed from time warping is always a multiple of the training speed. For example, if the training speed at any instance is 20mph, the resulting testing speed can only be multiples of 20mph such as 60mph, 40mph, 20mph or 10mph. Figure 3(b) plots the speed estimation from a warping path when the training speed is a constant at 20mph. The speed estimates from different segments of the warping path are depicted based on Type-1, Type-2, and Type-3 matches. Furthermore, the resulting speed estimates from the

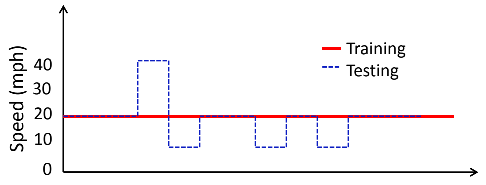


(a) Cost Matrix $C(M, N)$ and the warping path generation



Training
Testing
Speed of Testing = 2*Speed of Training
Speed of Testing = Speed of Training/2

(b) Speed estimation from a warping path



(c) Estimated speed from a warping path

Fig. 3. Illustration of vehicle speed estimation from DDTW.

testing trace versus the training speed are illustrated in Figure 3(c).

We observed that there are speed fluctuations from the estimated speed. In order to remove these fluctuations, we apply a moving window smoothing filter over the estimated speed which averages the speed estimates within the entire window to produce a single speed. The choice for the window size should not be too large since this might smooth out all variations leaving a very coarse speed estimate. Similarly, having a very small window size may result in the overall speed estimation to be highly fluctuating. We will evaluate the length of the optimal smoothing window in Section V.

IV. Slowdown Detection

In practice, most traffic engineering applications do not require the instantaneous speeds of vehicles and are more concerned about regions of bottlenecks. Such bottlenecks in road networks can in turn be detected from vehicular speeds by observing the regions where vehicles typically slowdown or by observing the normalized euclidean distance where the normalized euclidean distance between successive samples go below a threshold. We will next provide a formal definition of slowdown and describe the scheme for slowdown detection.

We define a slowdown as a sudden reduction in the speed of a moving vehicle by more than τ mph to a value below μ mph. The duration of the slowdown is the period of time the speed remains below μ mph. A slowdown is detected by sequentially scanning the input trace. The input trace can be the groundtruth speed data derived from GPS readings, DDTW estimated speed, speed estimate from the Localization algorithm, or the Normalized Euclidean Distance from Normalized Euclidean Distance algorithm. Our scheme identifies *peaks* and *dips* in the input trace. A peak occurs in the input trace at any given point when its first derivative (slope) at that point changes from positive to negative. Similarly dips occur when the slope changes from negative to positive. Our scheme initially assign a very low value to the first detected peak and a very high value to the first detected dip. As the scheme proceeds scanning the trace, the peak value is adjusted to the highest observed peak. Similarly, the dip value is adjusted to the lowest observed dip. After every adjustment of the peak and the dip, if $(peak - dip) > \tau$, and $dip < \mu$, a slowdown is declared. The duration of this slowdown is then the period of time the dip remains below μ. Finally, the peaks and dips are reset to the lowest and highest values respectively and the scheme repeats until all slowdowns are detected in the specific trace.

The main challenge in identifying slowdowns accurately lies on the choice of τ and μ. We performed an emperical study on 18 of our GPS traces that lasted for a total of 6.4 hours and picked a threshold of 25mph for τ since most breaking events involved slowding down the vehicle from 40-45mph speed limit in arterial roads to a very slow speed of around 5-10mph. Our choice for μ is 20mph because most residential regions have a speed limit of 25mph or more and we do not want to classify those residential regions as bottlenecks.

While a choice of 25mph and 20mph for τ and μ fits the ground truth speed from GPS, these thresholds need not be the same for the speeds estimated from either DDTW or Localization. and the normalized euclidean distance estimated by Normalized Euclidean Distance algorithm. For example, we showed in Section III-B that the speed estimate from DDTW at every instance is a multiple of the training speed which in turn requires a moving window smoothing filter to be applied over the estimated speed to get the speed estimate. However, due to this smoothing, an actual speed change of 25mph in the ground-truth speed may only correspond to a speed change of 15mph in the estimated speed.

In order to capture this relationship between the ground-truth speed and the estimates from other algorithms, we performed a regression analysis using a linear least square fit over the data sets. The inputs to the regression analysis are the ground-truth speed from GPS and the output from any of the three estimation algorithms under study: DDTW, Localization and Normalized Euclidean Distance. For instance, Figure 4 shows a scatter plot of the ground-truth speed from GPS versus the estimated speed from DDTW, and the corresponding fitted line obtained from linear least square fit. The slope of the fitted line determines that a drop in ground-truth speed by 25mph corresponds to only 15mph drop in the estimated speed from DDTW. Similarly, a μ value of 20mph in ground-truth corresponds to 21mph in estimated speed from DDTW. Table IV summarizes the values for τ and μ obtained using the above process for the different slowdown estimation algorithms.

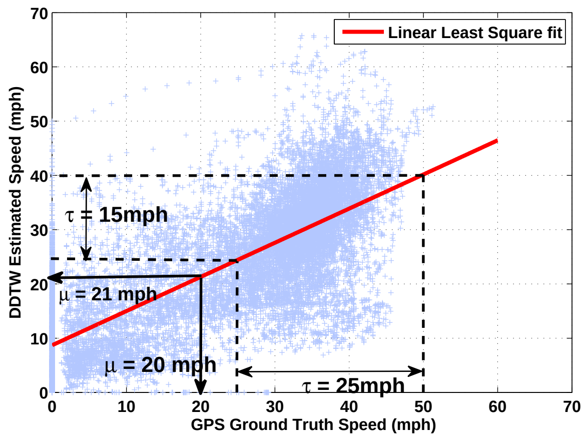


Fig. 4. Least square fit between the ground-truth speed from GPS and the estimated speed from DDTW.

The slowdown detection algorithm takes any of the four inputs, namely, *ground-truth speed trace*, *estimated speed trace from DDTW*, *estimated speed trace from Localization* or *normalized euclidean distance trace from Norm.Euc.Dist algorithm*, along with their respective τ andμ values. Figure 5 illustrates the results of slowdown detection performed on a 1000 seconds long ground-truth speed trace from GPS and the corresponding estimated speed from DDTW respectively. We treat the slowdown detection obtained from the GPS data as the ground truth. The y-axis on the left side corresponds to the speed of the traces, while the y-axis on the right represents the duration of each of the identified slowdown locations.

Finally, we will use three metrics: *True Positive*, *False Positive* and *False Negative* to quantify how well the the different algorithms detect bottlenecks in Section V. In the slowdown detection, true positives are the time periods when the ground-truth slowdowns coincide with the slowdowns estimated in the result of the algorithm under consideration, i.e.,DDTW or Localization, or Normalized Euclidean Distance. Wherease false positives occurs when slowdown is detected in our scheme under consideration but not in the ground-truth. False negative is when the slowdown exists in the ground-truth, however, is not detected by our scheme.

	τ	μ
Ground-Truth Speed from GPS	25 mph	20 mph
Estimated Speed (DDTW)	15.73 mph	21.28 mph
Estimated Speed (Localization)	4.92 mph	20.84 mph
Norm. Euc. Dist.(Norm. Euc. Distance Algorithm)	4.15 dBm	26 dBm

TABLE I
THRESHOLDS τ AND μ FOR THE SLOWDOWN ESTIMATION ALGORITHMS

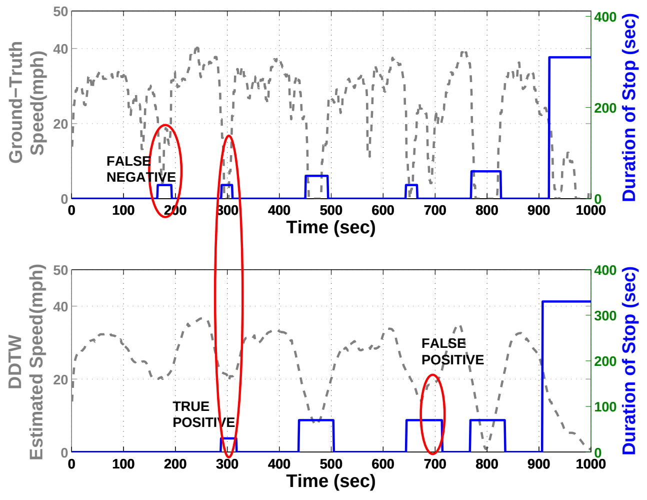


Fig. 5. Figure Illustrating the metrics for quantifying the slowdown detection performance, namely, false negative, true positive and false positive

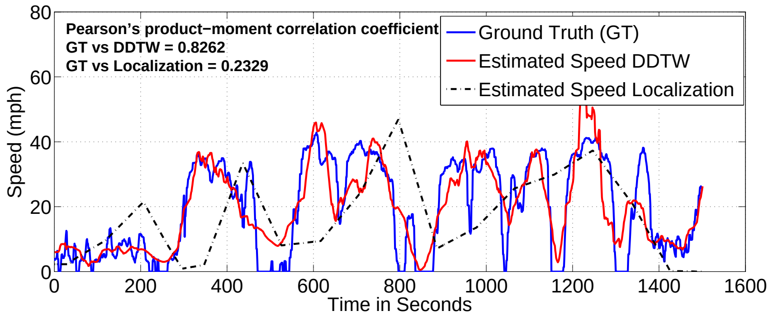
V. Evaluation

In this section, we evaluate the DDTW algorithm in terms of its ability to estimate instantaneous speed and detect slowdowns. We first describe our data collection methodology. Then, we present evaluation results for estimating instantaneous speed and detecting slowdowns on residential roads using GSM signal strength from mobile phones. Next, we study the sensitivity of DDTW to the alignment error. Finally, we show the generality of DDTW by presenting its speed estimation accuracy on a trace collected indoors with WiFi-based equipment.

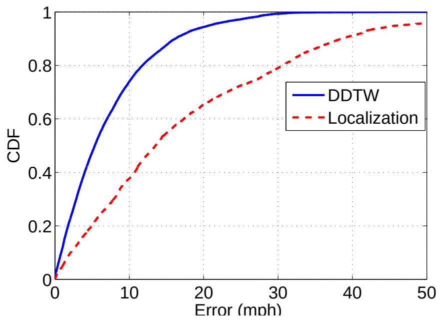
A. Data Collection

To evaluate DDTW, we collected two sets of data. The first set of data was collected outdoors using GSM enabled HTC Typhoon phones running the Intel-POLS [2] software. The software records the time, cell tower description (Cell ID, MNC, MCC, LAC, IMEI), and the received signal strength from the 7 strongest cells once every second. We used Holux GPSlim236 GPS receivers paired with mobile phones through Bluetooth for logging the ground truth location information for all traces. We collected 18 signal strength traces on a 10 mile long road, located in a residential area with lots of traffic lights. We chose the road with traffic lights to ensure the highly variable speed traffic pattern of our traces. We used one of the 18 traces as training and the remaining 17 traces as the testing traces. In total, the 17 testing traces contain 6.4 hours of data.

The second set of data was collected indoors on a 802.11b Wi-Fi network. We collected 9 traces in which the experimenter placed a laptop equipped with 802.11b WG511T Wi-Fi card in a cart and moved along a corridor measuring 228 ft in length thrice at three different speeds (1ft/sec, 2ft/sec, 4ft/sec) while sending out packets at the rate of 2 pkts/sec. Three receivers were placed along the corridor: one closer to the beginning of the corridor, one in the middle and one closer to the end of the corridor. The receivers were 802.11b/g



(a) Ground truth and estimated speeds of DDTW and Localization



(b) CDF of the speed estimation error

Fig. 6. Comparison of speed estimation accuracy of DDTW and Localization algorithms.

enabled, configured to listen on channel 6 in monitor mode and log the packets using Tshark [3] packet sniffer utility. In total, we had 550 seconds of data logged at each of the receivers.

Unless otherwise noted, we used the first set of data for all experiments in this paper.

B. Speed Estimation Accuracy

In this section, we evaluate the accuracy with which DDTW can estimate instantaneous speed and compare it to the accuracy achieved by the Localization algorithm, described in Section II. We do not include results for the Normalized Euclidean Distance algorithm in this section since the algorithm can be used to detect slowdowns only and cannot be used to estimate speed. Figure 6(a) plots the ground truth (actual) speed of a vehicle obtained through GPS, as well as the estimated speeds of the DDTW and the Localization algorithms. The drive took 1500 seconds to complete. The figure shows that the speed estimated by DDTW matches the actual speed very well, whereas the Localization algorithm performs poorly. To quantify how closely the two algorithms follow the actual speed, we calculated the Pearson's product-moment correlation coefficients between each of the algorithms and the actual speed. The Pearson's correlation coefficient measures a linear dependence between two variables. The coefficient of 1 means very strong positive correlation. The coefficient of 0 means no correlation. The Pearson's correlation coefficients are shown in the upper left corner of the figure. The actual speed and the DDTW algorithm exhibit very strong correlation of 0.83. The Localization algorithm, on the other hand, has a weak correlation with the actual speed of 0.34. We also calculated the Pearson's correlation coefficients for all the testing traces combined. The correlation between the estimated speed of DDTW and the actual speed was strong with a correlation coefficient of 0.75, whereas the correlation between the estimated speed of Localization and the actual speed is poor with a correlation coefficient of just 0.11.

Figure 6(b) plots the CDF of the error between the actual and the estimated speeds for both DDTW and Localization algorithms for all testing traces combined. The results show that DDTW performs significantly better than Localization, achieving the median error of 5.2 mph, which is more than twice lower than 13 mph achieved by the Localization algorithm. The median error of 5.2 mph is encouraging and we believe is suitable for a wide range of traffic engineering applications. In the next section, we will evaluate a method that can be used by an application that detects regions of bottleneck on a road. This requires detecting when a vehicle significantly reduces its speed.

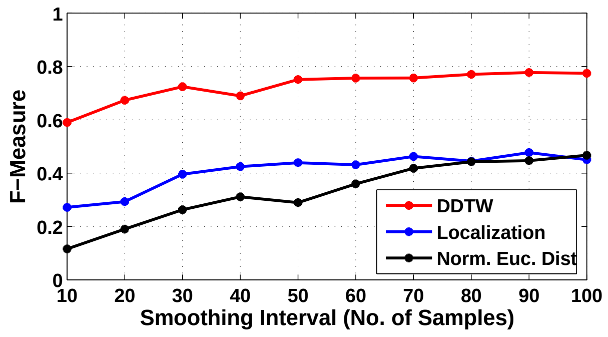


Fig. 7. The effect of the smoothing interval on the F-measure.

C. Slowdown Detection Accuracy

In this section, we evaluate the accuracy with which DDTW, Localization and Normalized Euclidean Distance algorithms can detect slowdowns, as defined in Section IV. Recall that a true positive occurs when an algorithm correctly predicts that there is a slowdown. A false positive occurs when an algorithm predicts that there is a slowdown but there is none. A false negative occurs when an algorithm doesn't predict a slowdown and there is one. Please refer to Figure 5 for an illustration of all these metrics.

We present our results using Precision, Recall and F-measure [18]. Precision captures the percentage of correct slowdown predictions and it is defined as the total duration of true positives divided by the sum of total duration of true positives and false positives. The higher is the Precision, the more accurate an algorithm's estimations are. Recall captures the percentage of actual slowdowns that were detected and it is defined as the total duration of true positives divided by the sum of true positives and false negatives. The higher the Recall, the more actual instances of a slowdown an algorithm has predicted correctly. F-measure is used to estimate the Precision/Recall tradeoff and it is defined as follows:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

The higher the F-measure, the better is an algorithm's slowdown detection accuracy. In our case, it is possible to trade off Precision for Recall by changing the smoothing interval, as defined in Section III-B. Figure 7 plots the F-measure for different smoothing intervals for the DDTW, Localization and Normalized Euclidean Distance algorithms.

The figure illustrates that the F-measure for DDTW is almost twice as high as the F-measure for Localization and Normalized Euclidean Distance algorithms across the entire range of smoothing intervals. This indicates that the DDTW has higher Precision and Recall compared to the other algorithms. We pick the smoothing interval that achieved the highest recall value for each algorithm, which in this case, was 50, 90 and 100 for DDTW, Localization and Norm. Euc. Dist respectively.

	Precision	Recall
DDTW	0.68	0.84
Localization	0.38	0.63
Normalized Euclidean Distance	0.39	0.59

TABLE II
SLOWDOWN DETECTION PERFORMANCE OF DDTW, LOCALIZATION AND NORMALIZED EUCLIDEAN DISTANCE ALGORITHMS.

Table II summarizes the Precision and Recall values for DDTW, Localization and Normalized Euclidean Distance algorithms for their respective optimal smoothing intervals derived from their F-Measure in Figure 7. DDTW significantly outperforms the other two algorithms achieving Precision of 0.68 and Recall of 0.84. Its Precision is 94% higher than that of Localization and 74% higher than that of Normalized Euclidean Distance. Its Recall is 40% higher than that of Localization and 42% higher than that of Normalized Euclidean Distance.

Next, we study the impact of the duration of a slowdown on the ability of the algorithms to detect it. The intuition tells that it should be easier to detect slowdowns of a longer duration. The duration of a slowdown is defined in Section IV as the total time the speed remains below the threshold of μ mph.

Figure 8 plots a histogram of the number of slowdowns of a given length that appear in the trace and the number of slowdowns that are correctly detected by each of the three algorithms. Although all algorithms can correctly detect all slowdowns of 120 seconds or more, only DDTW detects all slowdowns that are longer than 30 seconds. The algorithms cannot detect slowdowns of a short duration because of the smoothing that is applied to average out the oscillations in speed predictions, which in turn results in smoothing out abrupt speed changes that last for short durations.

D. Effect of Alignment Error on Speed Estimation Accuracy

In this section, we study the effect of the alignment error between the training and testing traces on the speed estimation accuracy of DDTW. Recall from Section III that introducing an alignment error results in applying DDTW on training and testing traces that are shifted in time by the value of the alignment error. Note that although we study the effect of alignment error of up to 500m, a typical GSM based localization system has a median localization error of less than 100m [5]. Therefore, it is reasonable to assume that, in practice, DDTW would achieve speed estimation accuracy that is equivalent to the one obtained with a 100m alignment error.

Alignment Error(m)	Median Error (mph)
0	5.2
100	6.5
200	7.12
500	8.57

TABLE III
EFFECT OF ALIGNMENT ERROR ON SPEED ESTIMATION ACCURACY

Table III summarizes the median error in miles per hour for different alignment errors. When a localization systems provides an accurate location estimate, DDTW suffers from no alignment errors and has a median speed estimation error of 5.2 mph. When an alignment error of 100m is present, the accuracy of DDTW degrades slightly to 6.5 mph. Even in this case, DDTW performs much better than the Localization algorithm that achieves the median speed estimation accuracy of 13 mph.

E. Indoor WiFi-based Experiment

We finally verify if DDTW technique can be used across a different wireless technology and a different environment for the same purpose of speed estimation. To this end, we use the Wi-Fi data in which we performed 9 indoor Wi-Fi experiments where the experimenter moved between the given two points in a long(228ft) corridor thrice at three different speeds, namely : 1ft/sec(0.68mph), 2ft/sec(1.36mph), 4ft/sec(2.72mph) while sending out packets at the rate of 2 pkts/sec. We had three Wi-Fi receivers, each recording the RSS from this transmitter.

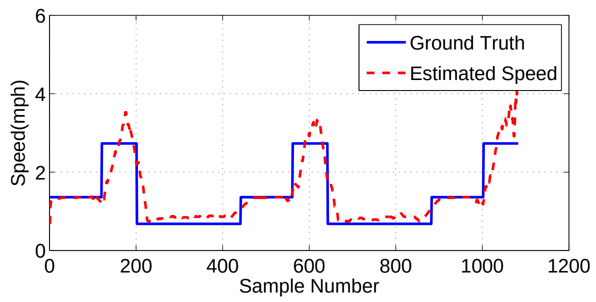
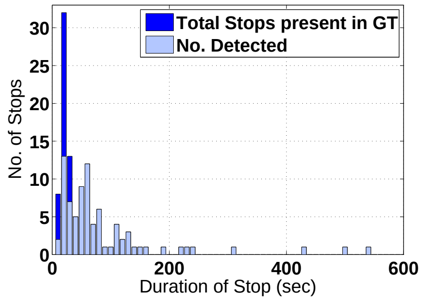
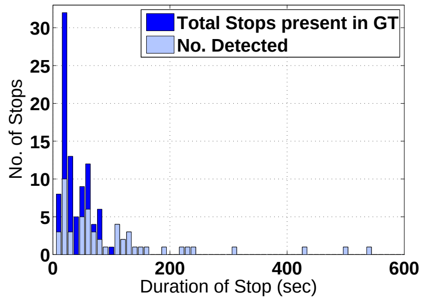


Fig. 9. Speed estimates from DDTW on indoor environment using RSS from receiver-1

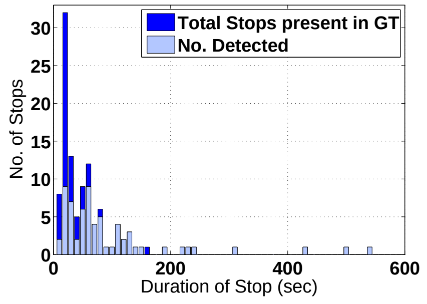
We can see from Figure 9 that the estimated speed closely follows the ground truth. The median error for this receiver is 0.1527mph. The median errors on receiver-2 and receiver-3 were 0.1388mph and 0.1527mph. This result is encouraging since this proves the generality of the proposed mechanism across different wireless technologies and shows that it is effective at detecting even very small speed changes indoors.



(a) DDTW



(b) Localization



(c) Normalized Euc Distance

Fig. 8. No. of slowdowns with different durations predicted by (a)DDTW (b)Localization (c)Normalized Euclidean Distance

VI. Conclusion

In this paper, we exploit received signal strength from mobile phones to track vehicular speed variations and predict bottlenecks on road segments. Our speed tracking mechanism is grounded on Derivative Dynamic Time Warping (DDTW) operating between traces of mobile phone signal strengths. Our approach makes use of the stability of signal strength measurements over time on any given road segment to optimally align the training and testing signal strength traces. Tracking of the speed variations, e.g., slowdowns, is then enabled based on the alignment produced by DDTW. We experimentally evaluated our mechanism on real signal strength measurements captured with mobile phones through various road drives and showed that our speed tracking has a very high correlation with the ground-truth speed reported by the GPS and exhibits a median error within $\pm\, 5mph$ across the 6.4 hours of driving traces. Moreover, our approach achieves a precision of 68% and a recall of 84% when predicting bottlenecks in road segments in terms of vehicular slowdowns. Additionally, to demonstrate the generality of our proposed speed tracking technique, we applied our approach on experimental traces from an indoor environment with walking people carrying Wi-Fi (802.11b) radios and succesfully showed the effectiveness of our mechanism across different environments and different radios.

References

[1] http://traffic.berkeley.edu/.

[2] Privacy observant location system (pols). http://pols.sourceforge.net/.

[3] Tshark - network protocol analyzer and traffic dumper. http://tinyurl.com/zc98g.

[4] G. Chandrasekaran, T. Vu, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. Vehicular speed estimation using received signal strength from mobile phones. In *UBICOMP*. ACM, Sep 2010.

[5] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *Ubicomp*, Lecture Notes in Computer Science, pages 225–242. Springer-Verlag, September 2006.

[6] B. Coifman. Using dual loop speed traps to identify detector errors. In *Transportation Research Board*, pages 47–58, 1999.

[7] B. Coifman. Improved velocity estimation using single loop detectors. In *Transportation Research Part A*, pages 863–880, December 2001.

[8] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *RATFG-RTS '01: Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, page 82, Washington, DC, USA, 2001. IEEE Computer Society.

[9] D. D. D.J and F. S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. In *Intelligent Transportation Systems*. IEEE, Jun 2000.

[10] D. Gundlegard and J. Karlsson. Handover location accuracy for travel time estimation in gsm and umts. In *IEEE Intelligent Transportation Systems Conference(ITSC)*, march 2009.

[11] J. Herrera and A.M.Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. 87^{th} *TRB Annual Meeting*, 2008.

[12] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys '08*, pages 15–28, New York, NY, USA, 2008. ACM.

[13] E. J and M. J. Pazzani. Derivative dynamic time warping. In *In First SIAM International Conference on Data Mining (SDM2001)*, 2001.

[14] T. Kahveci, A. Singh, and A. Grel. Similarity searching for multi-attribute sequences. In *In Proc. of SSDBM*, pages 175–184, 2002.

[15] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 28(6):623–635, 1980.

[16] V. Niennattrakul and C. A. Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. In *MUE '07: Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, pages 733–738, Washington, DC, USA, 2007. IEEE Computer Society.

[17] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In *Proceedings of the Seventh International Conference on Ubiquitous Computing (UbiComp 2005)*, September 2005.

[18] Rijsbergen and C. J. Van. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 1979.

[19] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:43–49, 1978.

[20] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. de Lara. Mobility detection using everyday gsm traces. In *UBICOMP*, 2006.

[21] S. Thajchayapong, W. Pattara-atikom, N. Chadil, and C. Mitrpant. Enhanced detection of road traffic congestion areas using cell dwell times. In *IEEE Intelligent Transportation Systems Conference(ITSC)*, Toronto, Ont., september 2006.

[22] A. Thiagarajan, L. R. Sivalingam, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan. Vtrack: Accurate, energy-aware traffic delay estimation using mobile phones. In *SenSys*, Berkeley, CA, November 2009.

[23] C. Xiao, K. D. Mann, and J. C. Olivier. Mobile speed estimation for tdma-based hierarchical cellular systems. In *IEEE VTC*, pages 2456–2460, Sept 1999.

[24] Y. Zhao. Standardization of mobile phone positioning for 3g systems. *IEEE Communications Magazine*, pages 108–116, 2002.

[25] Y. R. Zheng and C. Xiao. Mobile speed estimation for broadband wireless communications over rician fading channels. *Trans. Wireless. Comm.*, 8(1):1–5, 2009.