How Long to Wait?: Predicting Bus Arrival Time with Mobile Phone based Participatory Sensing

Pengfei Zhou, Yuanqing Zheng, Mo Li
Nanyang Technological University, Singapore
pfzhou@ntu.edu.sg, yuanqing1@e.ntu.edu.sg, limo@ntu.edu.sg

ABSTRACT

The bus arrival time is primary information to most city transport travelers. Excessively long waiting time at bus stops often discourages the travelers and makes them reluctant to take buses. In this paper, we present a bus arrival time prediction system based on bus passengers' participatory sensing. With commodity mobile phones, the bus passengers' surrounding environmental context is effectively collected and utilized to estimate the bus traveling routes and predict bus arrival time at various bus stops. The proposed system solely relies on the collaborative effort of the participating users and is independent from the bus operating companies, so it can be easily adopted to support universal bus service systems without requesting support from particular bus operating companies. Instead of referring to GPS enabled location information, we resort to more generally available and energy efficient sensing resources, including cell tower signals, movement statuses, audio recordings, etc., which bring less burden to the participatory party and encourage their participation. We develop a prototype system with different types of Android based mobile phones and comprehensively experiment over a 7 week period. The evaluation results suggest that the proposed system achieves outstanding prediction accuracy compared with those bus company initiated and GPS supported solutions. At the same time, the proposed solution is more generally available and energy friendly.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems – *Distributed Applications*; C.3.3 [**Special-Purpose and Application-based Systems**]: Real-time and embedded systems; H.5.3 [**Information Interfaces and Presentation**]: Groups and Organization Interfaces – *Collaborative Computing*

General Terms

Design, Implementation, Experimentation, Measurement, Performance

Keywords

Bus arrival time prediction, Participatory sensing, Mobile phones

1. INTRODUCTION

Public transport, especially the bus transport, has been well developed in many parts of the world. The bus transport services reduce the private car usage and fuel consumption, and alleviate traffic congestion. As one of the most comprehensive and affordable means of public transport, in 2011 the bus system serves over 3.3 million bus rides every day on average in Singapore with around 5 million residents [1].

When traveling with buses, the travelers usually want to know the accurate arrival time of the bus. Excessively long waiting time at bus stops may drive away the anxious travelers and make them reluctant to take buses. Nowadays, most bus operating companies have been providing their timetables on the web freely available for the travelers. The bus timetables, however, only provide very limited information (e.g., operating hours, time intervals, etc.), which are typically not timely updated. Other than those official timetables, many public services (e.g., Google Maps) are provided for travelers. Although such services offer useful information, they are far from satisfactory to the bus travelers. For example, the schedule of a bus may be delayed due to many unpredictable factors (e.g., traffic conditions, harsh weather situation, etc). The accurate arrival time of next bus will allow travelers to take alternative transport choices instead, and thus mitigate their anxiety and improve their experience. Towards this aim, many commercial bus information providers offer the realtime bus arrival time to the public [20]. Providing such services, however, usually requires the cooperation of the bus operating companies (e.g., installing special location tracking devices on the buses), and incurs substantial cost.

In this paper, we present a novel bus arrival time prediction system based on crowd-participatory sensing. We interviewed bus passengers on acquiring the bus arrival time. Most passengers indicate that they want to instantly track the arrival time of the next buses and they are willing to contribute their location information on buses to help to establish a system to estimate the arrival time at various bus stops for the community. This motivates us to design a crowd-participated service to bridge those who want to know bus arrival time (querying users) to those who are on the bus and able to share their instant bus route information (sharing users). To achieve such a goal, we let the bus passengers themselves cooperatively sense the bus route information using commodity mobile phones. In particular, the sharing passengers may anonymously upload their sensing data collected on buses to a processing server, which intelligently processes the data and distributes useful information to those querying users.

Our bus arrival time prediction system comprises three major components: (1) Sharing users: using commodity mobile phones as well as various build-in sensors to sense and report the lightweight cellular signals and the surrounding environment to a backend server; (2) Querying users: querying the bus arrival time for a particular bus route with mobile phones; (3) Backend server: collecting the instantly reported information from the sharing users, and intellectually processing such information so as to monitor the bus routes and predict the bus arrival time. No GPS or explicit location services are invoked to acquire physical location inputs.

Such a crowd-participated approach for bus arrival time prediction possesses the following several advantages compared with conventional approaches. First, through directly bridging the sharing and querying users in the participatory framework, we build our system independent of the bus operating companies or other third-party service providers, allowing easy and inexpensive adoption of the proposed approach over other application instances. Second, based on the commodity mobile phones, our system obviates the need for special hardware or extra vehicle devices, which substantially reduces the deployment cost. Compared with conventional approaches (e.g., GPS supported ones [12, 29]), our approach is less demanding and much more energy-friendly, encouraging a broader number of participating passengers. Third, through automatically detecting ambient environments and generating bus route related reports, our approach does not require the explicit human inputs from the participants, which facilitates the involvement of participatory parties.

Implementing such a participatory sensing based system, however, entails substantial challenges. (1) Bus detection: since the sharing users may travel with diverse means of transport, we need to first let their mobile phones accurately detect whether or not the current user is on a bus and automatically collect useful data only on the bus. Without accurate bus detection, mobile phones may collect irrelevant information to the bus routes, leading to unnecessary energy consumption or even inaccuracy in prediction results. (2) Bus classification: we need to carefully classify the bus route information from the mixed reports of participatory users. Without users' manual indication, such automatic classification is non-trivial. (3) Information assembling: One sharing user may not stay on one bus to collect adequate time period of information. Insufficient amount of uploaded information may result in inaccuracy in predicting the bus route. An effective information assembling strategy is required to solve the jigsaw puzzle of combining pieces of incomplete information from multiple users to picture the intact bus route status.

In this paper, we develop practical solutions to cope with such challenges. In particular, we extract unique identifiable fingerprints of public transit buses and utilize the microphone on mobile phones to detect the audio indication signals of bus IC card reader. We further leverage the accelerometer of the phone to distinguish the travel pattern of buses to other transport means. Thus we trigger the data collection and transmission only when necessary (§3.3). We let the mobile phone instantly sense and report the nearby celltower IDs. We then propose an efficient and robust top-k celltower set sequence matching method to classify the reported celltower sequences and associate with different bus routes. We intellectually identify passengers on the same bus and propose a celltower sequence concatenation approach to assemble their celltower sequences so as to improve the sequence matching accuracy (§3.4). Finally, based on accumulated information, we are then able to utilize both historical knowledge and the realtime traffic conditions to accurately predict the bus arrival time of various routes (§3.5).

We consolidate the above techniques and implement a prototype system with the Android platform using two types of mobile phones (Samsung Galaxy S2 i9100 and HTC Desire). Through our 7-week experimental study, the mobile phone scheme can accurately detect buses with 98% detection accuracy and classifies the bus routes with up to 90% accuracy. As a result, the prototype system predicts bus arrival time with average error around 80 seconds. Such a result is encouraging compared with current commercial bus information providers in Singapore.

In the following of this paper, we first introduce the background and motivation in §2. In §3, we detail the challenges of our system and describe our technical solutions. The evaluation results are presented in §4. The limitations and possible improvements are discussed in §5 followed by the description of related works in §6. We summarize this paper in §7.

2. BACKGROUND AND MOTIVATION

The bus companies usually provide free bus timetables on the web. Such bus timetables, however, only provide very limited information (e.g., operating hours, time intervals, etc.), which are typically not timely updated according to instant traffic conditions. Although many commercial bus information providers offer the realtime bus arrival information, the service usually comes with substantial cost. With a fleet of thousands of buses, the installment of in-vehicle GPS systems incurs tens of millions of dollars [29]. The network infrastructure to deliver the transit service raises the deployment cost even higher, which would eventually translate to increased expenditure of passengers.

For those reasons, current research works [12, 29] explore new approaches independent of bus companies to acquire transit information. The common rationale of such approaches is to continuously and accurately track the absolute physical location of the buses, which typically uses GPS for localization. Although many GPS-enabled mobile phones are available on the market, a good number of mobile phones are still shipped without GPS modules [31]. Those typical limitations of the localization based schemes motivate alternative approaches without using GPS signal or other localization methods. Besides, GPS module consumes substantial amount of energy, significantly reducing the lifetime of power-constrained mobile phones [31]. Due to the high power consumption, many mobile phone users usually turn

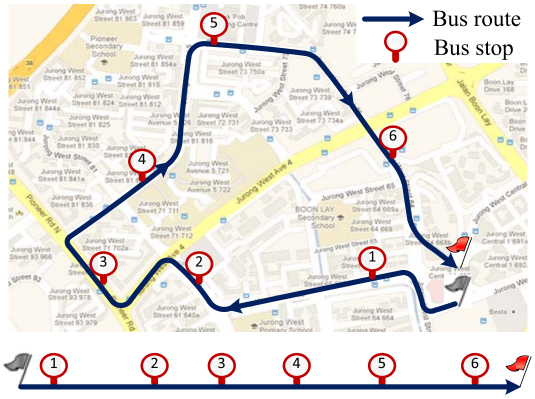


Figure 1: Absolute localization is unnecessary for arrival time prediction

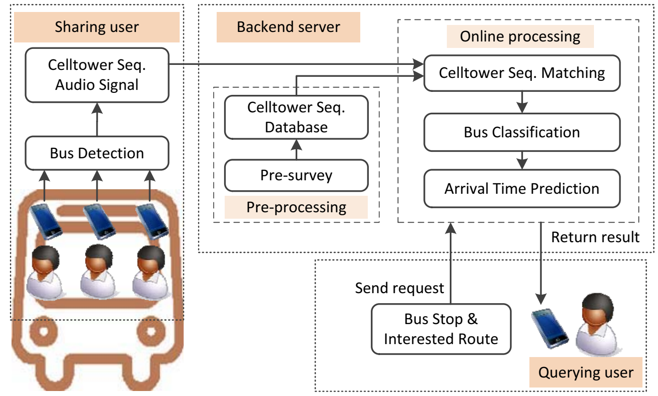


Figure 2: System architecture

off GPS modules to save battery power. The mobile phones in vehicles may perform poorly when they are placed without line-of-sight paths to GPS satellites [9].

To fill this gap, we propose to implement a crowd-participated bus arrival time prediction system utilizing cellular signals. Independent of any bus companies, the system bridges the gap between the querying users who want to know the bus arrival time to the sharing users willing to offer them real-time bus information. Unifying the participatory users, our design aims to realize the common welfare of the passengers.

To encourage more participants, no explicit location services are invoked so as to save the requirement of special hardware support for localization. Compared with the high energy consumption of GPS modules, the marginal energy consumption of collecting celltower signals is negligible on mobile phones. Our system therefore utilizes the celltower signals without reducing battery lifetime on sharing passengers' mobile phones. Our design obviate the need for accurate bus localization. As a matter of fact, since the public transport buses travel on certain bus routes (1D routes on 2D space), the knowledge of the current position on the route (1D knowledge) and the average velocity of the bus suffices to predict its arrival time at a bus stop. As shown in Figure 1, for instance, say the bus is currently at bus stop 1, and a querying user wants to know its arrival time at bus stop 6. Accurate prediction of the arrival time requires the distance between bus stop 1 and 6 along the 1D bus route (but not on the 2D map) and the average velocity of the bus. In general, the physical positions of the bus and the bus route on the 2D maps are not strictly necessary. In our system, instead of pursuing the accurate 2D physical locations, we logically map the bus routes to a space featured by sequences of nearby cellular towers. We classify and track the bus statuses in such a logical space so as to predict the bus arrival time on the real routes.

We leverage various lightweight sensors (e.g., microphone, accelerometer, etc.) on mobile phones to enable automatic and intelligent data collection and transmission. Although we can make use of a basket of instantly available sensor resources (e.g., magnetometer, gyroscope, camera, proximity sensors, etc.), we mainly focus on energy-friendly and widely available sensing signals (e.g., celltower and audio signals). The purpose is to make the solution lightweight and pervasively available to attract more participants.

3. SYSTEM DESIGN

Though the idea is intuitive, the design of such a system in practice entails substantial challenges. In this section, we describe the major components of the system design. We illustrate the challenges in the design and implementation, and present several techniques to cope with them.

3.1 System overview

Figure 2 sketches the architecture of our system. There are 3 major components.

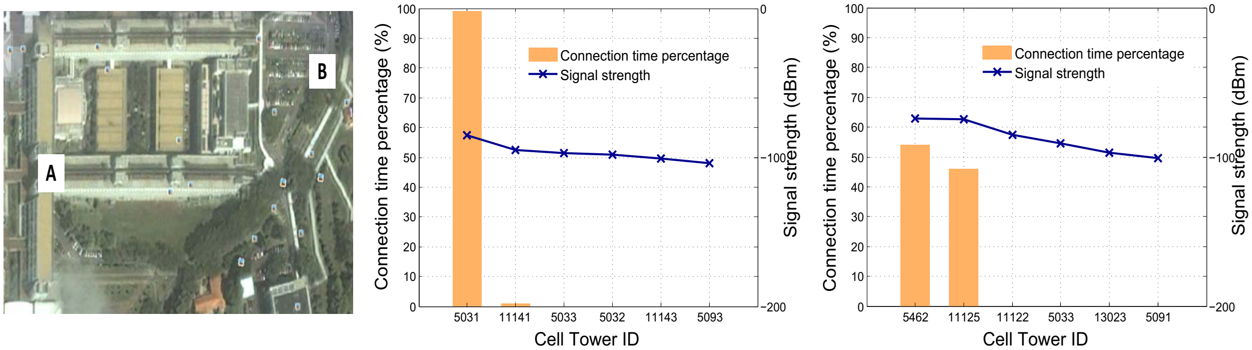
Querying user. As depicted in Figure 2 (right bottom), a querying user queries the bus arrival time by sending the request to the backend server. The querying user indicates the interest bus route and bus stop to receive the predicted bus arrival time.

Sharing user. The sharing user on the other hand contributes the mobile phone sensing information to the system. After a sharing user gets on a bus, the data collection module starts to collect a sequence of nearby celltower IDs. The collected data is transmitted to the server via cellular networks. Since the sharing user may travel with different means of transport, the mobile phone needs to first detect whether the current user is on a bus or not. As shown in Figure 2 (left side), the mobile phone periodically samples the surrounding environment and extracts identifiable features of transit buses. Once the mobile phone confirms it is on the bus, it starts sampling the celltower sequences and sends the sequences to the backend server. Ideally, the mobile phone of the sharing user automatically performs the data collection and transmission without the manual input from the sharing user.

Backend server. We shift most of the computation burden to the backend server where the uploaded information from sharing users is processed and the requests from querying users are addressed. Two stages are involved in this component.

In order to bootstrap the system, we need to survey the corresponding bus routes in the offline pre-processing stage. We construct a basic database that associates particular bus routes to celltower sequence signatures. Since we do not require the absolute physical location reference, we mainly wardrive the bus routes and record the sequences of observed celltower IDs, which significantly reduces the initial construction overhead.

The backend server processes the celltower sequences and



(a) Celltower coverage (b) Connection at position A (c) Connection at position B

Figure 3: Celltower connection time and received signal strength

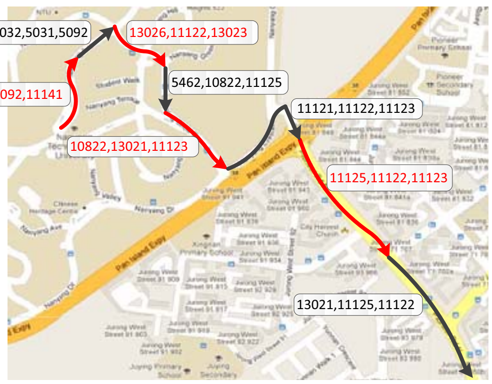


Figure 4: Celltower sequence set along a bus route

(a) On buses (b) At rapid train station entrances

Figure 5: Transit IC card readers

audio signals from sharing users on the buses in the online processing stage. Receiving the uploaded information, the backend server first distinguishes the bus route that the sharing user is currently traveling with. The backend server classifies the uploaded bus routes primarily with the reported celltower sequence information. The bus arrival time on various bus stops is then derived based on the current bus route statuses.

3.2 Pre-processing celltower data

The backend server needs to maintain a database that stores sequences of celltower IDs that are experienced along different bus routes. Wardriving along one bus route, the mobile phone normally captures several celltower signals at one time, and connects to the celltower with the strongest signal strength. We find in our experiments that even if a passenger travels by the same place, the connected celltower might be different from time to time due to varying celltower signal strength. To improve the robustness of our system, instead of using the associated celltower, we record a set of celltower IDs that the mobile phone can detect. To validate such a point, we do an initial experiment. We measure the celltower coverage at two positions A and B within the university campus, which are approximately 300 meters apart (Figure 3(a) depicts the two positions on the map).

Figure 3(b) and 3(c) report the celltower that the mobile phone can detect, as well as their average signal strength and connection time at A and B, respectively. We find that position A and position B are both covered by 6 celltowers with divergent signal strength. In Figure 3(b), we find that at position A the mobile phone is connected to the celltower 5031 over 99% of the time, while its signal strength remains consistently the strongest during the 10-hour measurement. In Figure 3(c), the mobile phone at position B observes two celltowers with comparable signal strength. We find that the mobile phone is more likely to connect to the celltower with stronger signal strength, and also may connect to the celltower with the second strongest signal strength. Nevertheless, during our 7-week experiments, we consistently observe that mobile phones almost always connect to the top-*3* strongest celltowers. Therefore, in practice we choose the set of the top-*3* strongest celltowers as the signature for route segments.

Figure 4 illustrates the celltower sequence collected on our campus bus traveling from our school to a rapid train station off the campus. The whole route of the bus is divided into several concatenated sub-route segments according to the change of the top-*3* celltower set. They are marked alternately in red and black in the figure. For example, the mobile phone initially connects to celltower 5031 in the first sub-route and the top-*3* celltower set is {5031, 5092, 11141}. Later the mobile phone is handed over to celltower 5032 and the celltower set becomes {5032, 5031, 5092} in the second sub-route. We subsequently record the top-*3* celltower in each sub-route.

Such a sequence of celltower ID sets identifies a bus route in our database. By wardriving along different bus routes, we can easily construct a database of celltower sequences associated to particular bus routes.

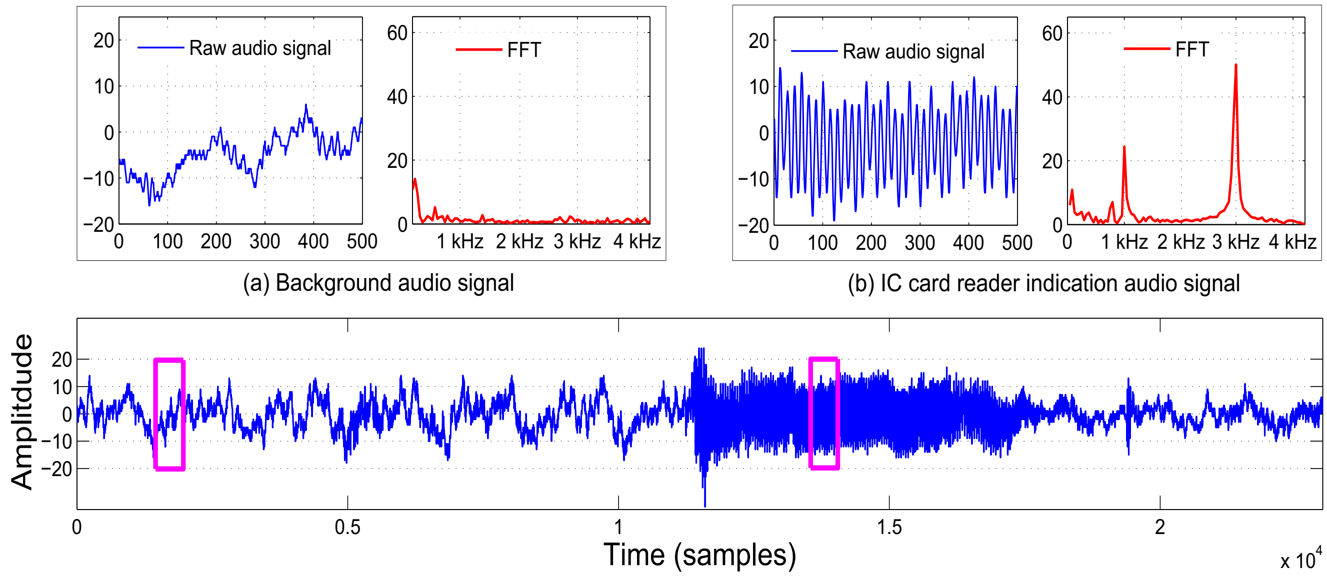


Figure 6: Bus detection using audio indication signal

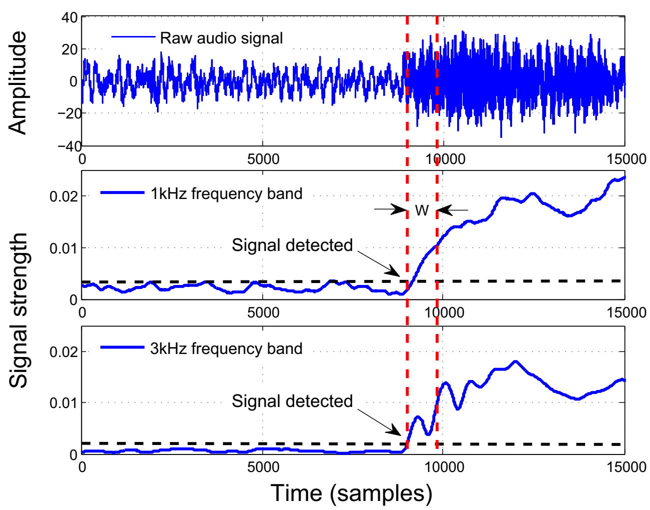


Figure 7: Detecting audio beeps in the frequency domain

3.3 Bus detection: Am I on a bus?

During the on-line processing stage, we use the mobile phones of sharing passengers on the bus to record the cell-tower sequences and transmit the data to the backend server. As aforementioned, the mobile phone should intelligently detect whether it is on a public transit bus or not and start to collect the data only when the mobile phone is on a bus. Some works [18, 21] study the problem of activity recognition and context awareness using various sensors. Such approaches, however, cannot be used to distinguish different transport modes (e.g., public transit buses and non-public buses). In this section, we explore multi-sensing resources to detect the bus environment and distinguish it from other transport modes. We seek a lightweight detection approach in terms of both energy consumption and computation complexity.

3.3.1 Audio detection

Nowadays, IC cards are commonly used for paying transit fees in many areas (e.g., EZ-Link cards in Singapore [2], Octopus cards in Hong Kong [3], Oyster cards in London [4], etc). On a public bus in Singapore, several card readers are deployed for collecting the fees (as depicted in Figure 5(a)). When a passenger taps the transit card on the reader, the reader will send a short beep audio response to indicate the successful payment. In our system, we choose to let the mobile phone detect the beep audio response of the card reader, since such distinct beeps are not widely used in other means of transportation such as non-public buses and taxis.

In order to exploit the unique beeps of IC card readers, in our initial experiment we record an audio clip on the bus at the audio sampling rate of 44.1kHz with Samsung Galaxy S2 i9100 mobile phone. Such a sampling rate is more than sufficient to capture the beep signals [26]. Figure 6 (bottom) plots the raw audio signal in the time domain, where the IC card reader starts beeping approximately from $11000th$ sample and lasts to $18000th$ sample. We crop the section of the beep audio signal and depict the section in Figure 6(b). After we convert the time domain signal to the frequency domain through 512pt Fast Fourier Transform (FFT) (Figure 6(b)), we observe clear peaks at 1kHz and 3kHz frequency bands. For comparison we depict the audio clip as well where no beep signal is sent. Both time domain and the frequency domain signals are plotted in Figure 6(a). We find no peaks at 1kHz and 3kHz frequency bands.

With the knowledge of the frequency range of the dual-tone beep signal sent by the IC card reader, in our system we can lower down the audio sampling rate of the mobile phone to 8kHz (8000 samples/s) which is sufficient to capture the beep signals with maximum frequency of 3kHz. We find that in practice 128pt FFT suffices to detect the IC card reader on the bus with tractable computation complexity on commodity mobile phones. We use the standard sliding window averaging technique with window size $w = 32$ samples to filter out the noises in both 1kHz and 3kHz frequency

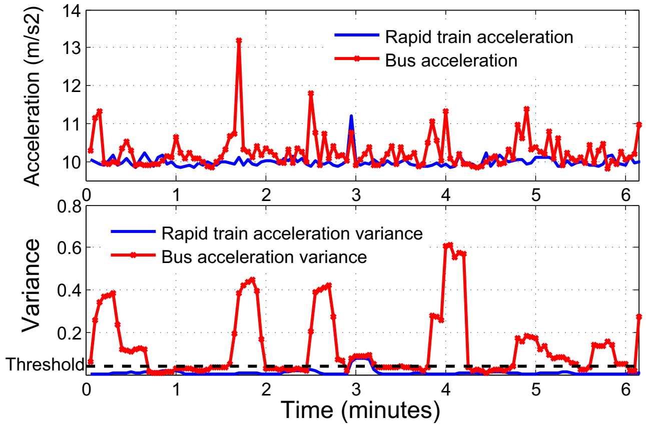


Figure 8: Accelerometer readings on rapid train and bus

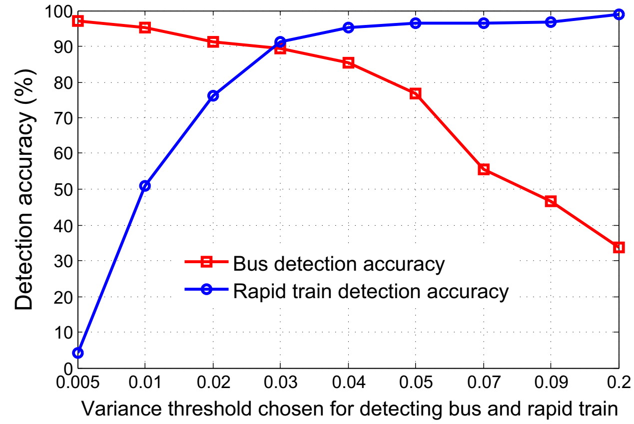


Figure 9: Variance threshold to distinguish rapid trains from buses

bands. We use an empirical threshold of three standard deviation (i.e., 99.7% confidence level of noise) to detect beep signals. If the received audio signal strengths in 1kHz and 3kHz frequency bands both exceed the threshold, the mobile phone confirms the detection of the bus. Figure 7 depicts the beep signal detection process. When the IC card reader starts beeping, the signal strengths in both 1kHz and 3kHz frequency bands jump significantly and therefore can be detected.

We test the audio indication based bus detection method with various scenarios, and the experiments show encouraging results for bus detection (§4.2.1). As the dual-tone responsive signal is universally used in almost all public transit buses in Singapore, we can use it as an identifiable signature to distinguish the buses from other vehicles. Therefore, we use the dual-tone as the acoustic trigger for the successive celltower data collection and transmission of the mobile phones of sharing users. We can easily adopt similar techniques [22] to detect certain audio indications to identify the public transports as well in other areas (e.g., the bell ringing tunes in Hong Kong buses).

3.3.2 Accelerometer detection: Bus v.s. Rapid train

In Singapore, however, transit IC cards are used in rapid train stations as well where the IC card readers in the entrances may send the same beep audio signal (Figure 5(b)). In practice, we find that solely relying on the audio detection the mobile phones may falsely trigger the celltower ID collection when they go with the rapid trains. Since the train routes have substantial above-ground segments that overlap with bus routes, simply using celltower signals does not effectively differentiate the two transit means. We expect to leverage the accelerometer sensor on the mobile phone to reduce such false detection.

Intuitively, the rapid trains are moving at relatively stable speeds with few abrupt stops or sharp turns. On the contrary, the buses are typically moving with many sharp turns and frequent acceleration and deceleration. We collect the accelerometer data at a moderate sampling rate of 20Hz. The raw accelerometer readings are first made orientation-independent by computing the L_2-norm (or magnitude) of the raw data [28]. Figure 8 (top) plots the accelerometer readings on a rapid train and a public transit bus which suggest that the accelerometer reading on the bus fluctuates much frequently with larger magnitudes. We explore such features of accelerometer readings to distinguish the buses from the rapid trains.

We measure the statistics of the accelerometer readings during 12.5 seconds (250 samples) to reduce the impact of noise, such as average and variance of the acceleration. Figure 8 (bottom) plots the variance of the accelerometer readings on the rapid train and the public transit bus, respectively. According to the figure, the variance on the bus is significantly larger than that on the train. Therefore, we distinguish the buses from the trains using the variance of accelerometer readings by setting a proper threshold.

We confirm the detection of buses if the measured acceleration variance is above the threshold, and the detection of rapid trains otherwise. In Figure 9, we vary the threshold from 0.005 to 0.2 and plot the detection accuracy. If the threshold is small, most buses will be correctly detected, while many trains will be misdetected as buses as well, which may lead to noisy inputs to the backend server and energy waste of mobile phones in collecting celltower IDs. On the other hand, if threshold is too big, most rapid trains will be filtered out, while we will miss the detection of many actual buses, which may lose the opportunities in collecting useful celltower information on the buses. We select an empirical threshold 0.03 to balance the false negative and false positive.

In practice, we find that accelerometer based detection can distinguish the buses from the trains with an accuracy of approximately 90% (§4.2.2). The error rate of falsely detecting rapid trains as buses is even smaller. The detection error of falsely classifying public buses into rapid trains is mainly due to the abnormality of the bus routes (e.g., long straight routes) especially during non-peak hours. Such a detection error is tolerable in the bus classification stage, where the backend server has information redundancy to handle the noisy reports.

3.4 Bus classification

When a sharing user gets on the bus, the mobile phone samples a sequence of celltower IDs and reports the information to the backend server. The backend server aggregates the inputs from massive mobile phones and classifies the in-

Figure 10: Celltower sequence matching

Database seq.	1 2 4	7 8 4 5	9 6
Uploaded seq.		7 8 4 5	
Matched seq.		7 8 4 5	

Table 1: Celltower sequence matching

puts into different bus routes. The statuses of the bus routes are then updated accordingly.

3.4.1 Celltower sequence matching

We match the received celltower sequences to those signature sequences store in the database. Figure 10 shows an illustrative example where a sharing passenger gets on the bus at location A. The backend server will receive a celltower sequence of $\langle 7, 8, 4, 5 \rangle$ when the sharing user reaches location B. Say that the celltower sequence of the bus route stored in the database is $\langle 1, 2, 4, 7, 8, 4, 5, 9, 6 \rangle$, then the sequence $\langle 7, 8, 4, 5 \rangle$ matches the particular bus route as a sub-segment as shown in Table 1.

In practical scenarios, the sequence matching problem becomes more complicated due to the varying celltower signal strength. Recall that for each sub-route we record the top-*3* celltower IDs instead of the connected celltower ID in the pre-processing period. We let each mobile phone send back the sequence of celltowers that the mobile phone has connected to. In the matching process on the server, we accordingly devise a top-k celltower sequence matching scheme by modifying the Smith-Waterman algorithm [33]. Smith-Waterman is a dynamic programming algorithm for performing local sequence alignment which has been widely used in bioscience (e.g., to determine similar regions between two nucleotide or protein sequences).

We make concrete modifications on the original algorithm to support the top-k celltower sequence matching. We weigh a matching of a celltower ID with a top-k set according to the celltower signal strength. Say that in a top-k set $S = \{c_1, c_2, \ldots, c_k\}$ ordered by signal strength (i.e., $s_i \geq s_j, 1 \leq i \leq j \leq k$), where c_i and s_i denote celltower i and its signal strength, respectively.

We denote the uploaded celltower sequence from a sharing user as $Seq_{\texttt{upload}} = \langle u_1 u_2 \ldots u_m \rangle$ where m is the sequence length. We also denote a celltower set sequence in database as $Seq_{\texttt{database}} = \langle S_1 S_2 \ldots S_n \rangle$ where n is the set sequence length. If $u_i = c_w \in S_j$, u_i and S_j are considered matching with each other, and mismatching otherwise. We assign a score $f(s_w)$ for a match, where $f(s_w)$ is a positive non-decreasing scoring function and w is the rank of signal strength. In practice, we use $f(s_w) = 0.5^{w-1}$ as the scoring function according to the signal strength order in the set. The penalty cost for mismatches is set to be an empirical value of -0.5 which balances the robustness and accuracy in practice.

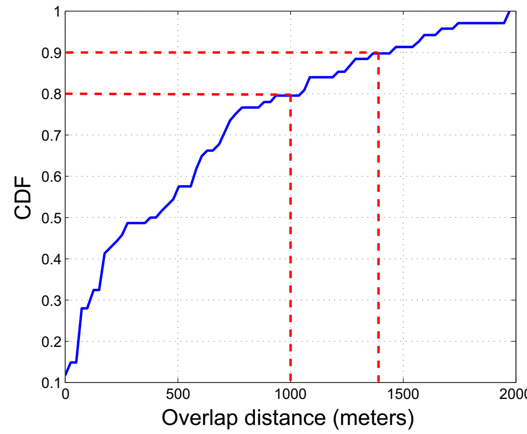


Figure 11: CDF of the overlapped route length

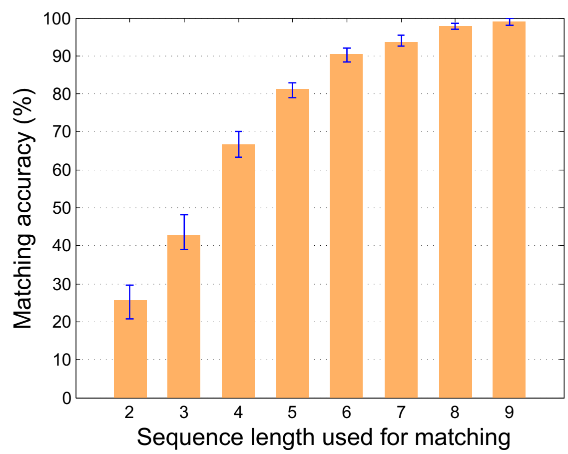


Figure 12: Matching accuracy with varying sequence length

We choose top-*3* celltower IDs with strongest celltower signal strength to form a set based on our initial observations (§3.2). The distinctive advantage of the proposed classification algorithm is its robustness to the variation of celltower signal strength. Table 2 shows a celltower set sequence matching instance. In the example, the uploaded celltower sequence is $Seq_{\texttt{upload}} = \langle 1, 8, 10, 15, 16 \rangle$, and the celltower ID set is shown in the first three rows sorted in decreasing order of the associated celltower signal strength.

After running the sequence matching algorithm across all bus route sequences in the database, the backend server selects the bus route with the highest score. If the highest matching score is smaller or the sequence length is shorter than our empirical thresholds, the backend server postpones

Database celltower set seq.	19	**1**	4	7	**10**	13	**16**	22	$\sum$
	20	2	5	**8**	11	14	17	23	
	21	3	6	9	12	**15**	18	24	
Uploaded seq.		1	–	8	10	15	16		
Score	0	+1	-0.5	+0.5	+1	+0.25	+1	0	3.25

Table 2: Top-*3* set sequence matching

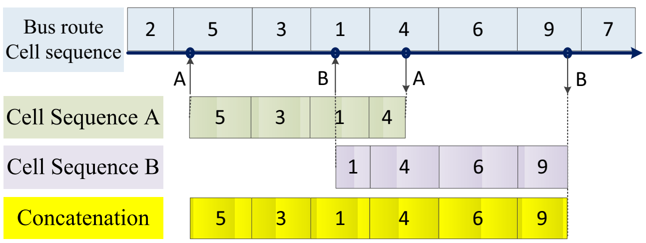


Figure 13: Celltower sequence concatenation

the updates to avoid errors. Intuitively, the small highest matching score would be due to mistriggering of sharing phones uploading celltower sequence not from interested bus routes (e.g., rapid trains, private cars, etc). Too short celltower sequence may not be informative since the misclassification rate of such short sequence is high and thus the backend server postpones the classification and the updating process until the sequence excesses the empirical threshold (which will be elaborated later).

One problem of the celltower sequence matching is that some bus routes may overlap with each other. The mobile phones on the overlapped road segments are likely to observe similar celltower sequences. Since many buses typically arrive at and depart from several major transit centers, such overlapping road segments among different bus routes are common.

We survey 50 bus routes in Singapore and measure their overlapped road segments using Google Maps. Figure 11 plots the distribution of the lengths of overlapped road segments, which suggests that over 90% of the overlapped route segments are shorter than 1400 meters, and over 80% are less than 1000 meters. Considering that the coverage range of each celltower in urban area is about 300-900 meters, we set the empirical threshold of celltower sequence length to 7.

Figure 12 plots the celltower sequence matching accuracy in classifying the bus routes. We vary the length of uploaded celltower sequence from 2 to 9. We find that the matching accuracy is low when the celltower sequence length is small (e.g, <4) largely because of the problem of route overlap. We observe that when the celltower sequence length reaches 6, the accuracy increases substantially to around 90%. When the celltower sequence length is larger than 8, the experimental results are reasonably accurate and robust.

3.4.2 *Celltower sequence concatenation: Solving jigsaw puzzles*

In many practical scenarios, the length of the celltower sequence obtained by a single sharing user, however, may be insufficient for accurate bus route classification. An intuitive idea is that we can concatenate several celltower sequences

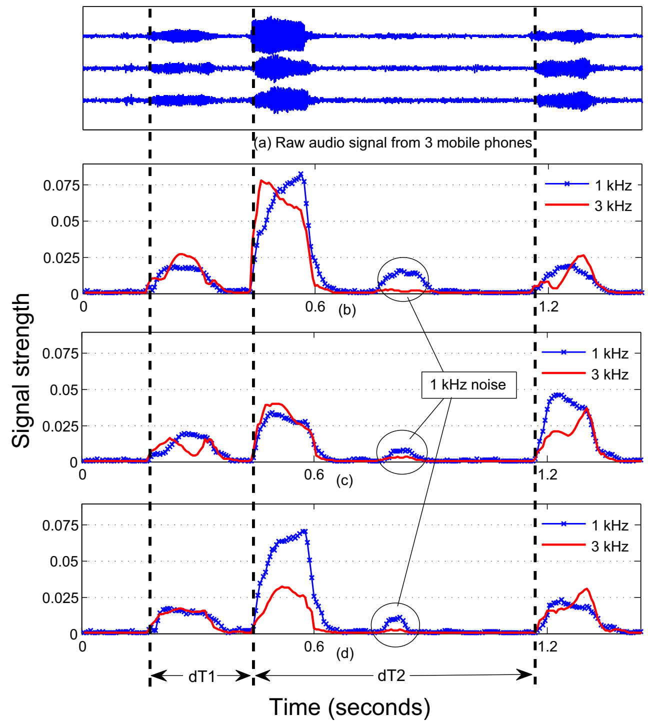


Figure 14: Time intervals of audio indication signals

of different sharing users on the same bus to form a longer celltower sequence. In Figure 13, both celltower sequences of sharing user A and B are short, while by concatenating the two celltower sequences the backend server may obtain an adequately long celltower sequence which can be used for more accurate bus classification. A simple way of concatenating the celltower sequences is to let the mobile phones of sharing passengers locally communicate with each other (e.g., over Bluetooth) [24]. This approach, however, mandates location exposure among sharing passengers and might raise privacy concerns. We thereby shift such a job to the backend server.

Recall that the mobile phone needs to collect audio signals for bus detection (§3.3.1). Here, we reuse such information to detect whether the sharing passengers are on the same bus for celltower sequence concatenation. At each bus stop, normally several passengers enter a bus and multiple beeps of the IC card readers can be detected. The time intervals between the consecutive beep signals fingerprint each bus in the time domain. Figure 14 depicts an instance of the audio signals captured by three different mobile phones on the same bus. We depict the raw audio signals in Figure 14(a), and corresponding frequency domain signals in Figure 14(b)-(d). Compared with the time domain signal, the frequency domain signal is robust against the background noise (e.g., though signal strength increases are observed in 1kHz frequency band around 0.8s, the signal strengths in 3kHz frequency band remain low). We can see that in the frequency domain the signals are highly cross-correlated and thus can be used to determine whether the phones are on the same bus. Specifically, the time intervals observed by three mobile phones are all approximately dT1 and dT2 in Figure 14.

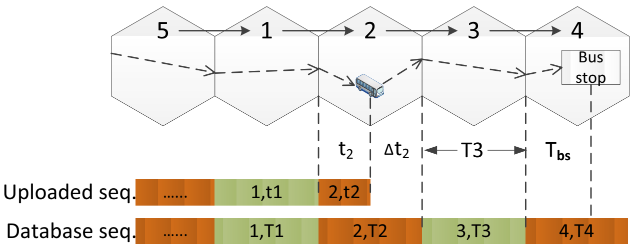


Figure 15: Bus arrival time prediction

We therefore use the time intervals between the detected beeps to determine whether multiple mobile phones are on the same bus. In our system, the mobile phones of sharing users keep sampling the audio signal and record the time intervals between the detected beeps. Such beep interval information is reported along with the celltower sequences to the backend server. Receiving the uploaded sensing data from sharing passengers, the backend server detects and groups the sharing passengers on the same bus by comparing both celltower sequences and the time intervals of the beep signals. The backend server concatenates the pieces of celltower sequences from the same bus and forms a longer celltower sequence.

3.5 Arrival time prediction

After the celltower sequence matching, the backend server classifies the uploaded information according to different bus routes. When receiving the request from querying users the backend server looks up the latest bus route status, and calculates the arrival time at the particular bus stop.

Figure 15 illustrates the calculation of bus arrival time prediction. The server needs to estimate the time for the bus to travel from its current location to the queried bus stop. Suppose that the sharing user on the bus is in the coverage of celltower 2, the backend server estimates its arrival time at the bus stop according to both historical data as well as the latest bus route status. The server first computes the dwelling time of the bus at the current cell (i.e., cell 2 in this example) denoted as t_2. The server also computes the traveling time of the bus in the cell that the bus stop is located denoted as t_{bs}. The historical dwelling time of the bus at cell 3 is denoted as T_3. The arrival time of the bus at the queried stop is then estimated as follows,

$$T = T_2 - t_2 + T_3 + t_{bs}$$

Without loss of generality, we denote the dwelling time in cell i as T_i, $1 \leq i \leq n$, the bus's current cell number as k, and the queried bus stop's cell number as q. The server can estimate the arrival time of the bus as follows,

$$T = \sum_{i=k}^{q-1} T_i - t_k + t_q$$

The server periodically updates the prediction time according to the latest route report from the sharing users and repsonds to querying users. The querying users may indicate desired updating rates and the numbers of successive bus runs to receive the timely updates.

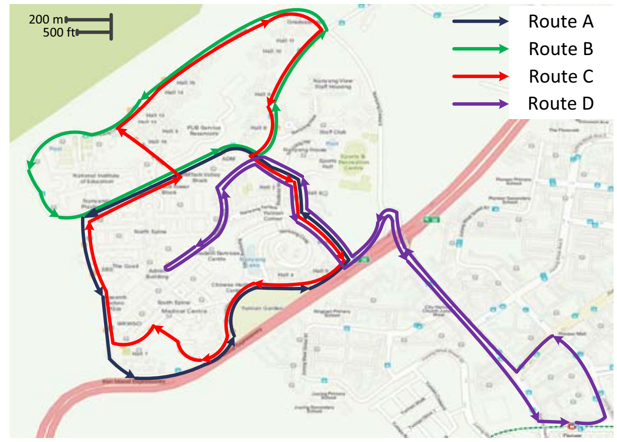


Figure 16: Campus shuttle bus routes

Route	Length	Avg. vel.	Stop	Seq. Length
A	4.0km	22.1km/h	11	14-15
B	3.8km	21.2km/h	9	9-10
C	5.5km	20.6km/h	13	16-17
D	5.8km	18.3km/h	9	20-22

Table 3: Campus bus route length, average velocity, number of bus stops, and celltower sequence length

Route	A	B	C	D
A	–	1.4km	3.4km	1.9km
B	1.4km	–	2.1km	0km
C	3.4km	2.1km	–	1.9km
D	1.9km	0km	1.9km	–

Table 4: The lengths of shared bus routes

4. IMPLEMENTATION AND EVALUATION

We implement a prototype system on the Android platform with different types of mobile phones, and collect the real data over a 7-week period. We first present the experiment environment and methodology (§4.1). We test and evaluate each component (bus detection in §4.2, and bus classification in §4.3) and present the overall performance of bus arrival time prediction in §4.4. The following details the experiment methodology and findings.

4.1 Experimental methodology

Mobile phones. We implement the mobile phone applications with the Android platform using Samsung Galaxy S2 i9100 and HTC Desire. Both types of mobile phones are equipped with accelerometers and support 16-bit 44.1kHz audio signal sampling from microphones. The Samsung Galaxy S2 i9100 has a 1GB RAM and Dual-core 1.2GHz Cortex-A9 processor, while the HTC Desire has a 768MB RAM and 1GHz Scorpion processor. For most of our experiments, we base on the SingTel GSM networks in Singapore.

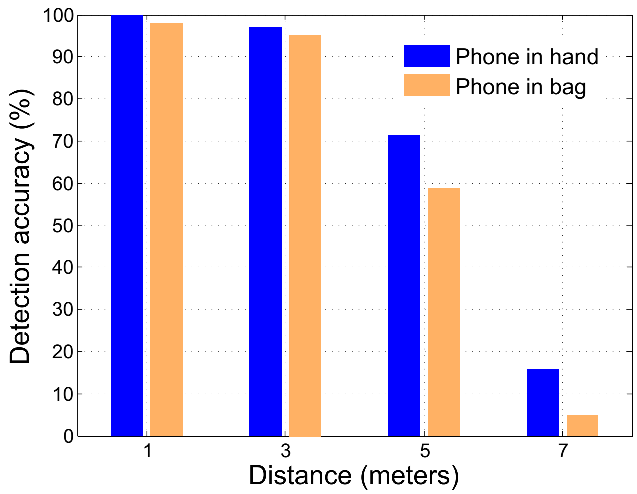


Figure 17: Bus detection accuracy

Scenario	DR	FPR	Accuracy
Mobile phone in hand			
1m	100%	3%	98%
3m	97%		97%
5m	71%		84%
7m	15%		56%
Mobile phone in bag			
1m	98%	1%	98%
3m	95%		97%
5m	59%		79%
7m	5%		52%

Table 5: Bus detection accuracy. Detection rate (DR), false positive rate (FPR) and accuracy under various scenarios

Backend server. We implement the backend server in Java running on the DELL Precision T3500 workstation with 4GB memory and Intel Xeon W3540 processor. The bus arrival time prediction service can be implemented in a computing cloud for dynamic and scalable resource provisioning as well [15].

Experiment environment. Public bus transit system serves millions of bus rides every day covering most parts of Singapore. The public bus transit system is supervised by Land Transport Authority (LTA) of Singapore and is commercially operated mainly by two major public transport providers, SBS Transit and SMRT Corporation [5, 20]. Many other transit means coexist with the public bus system. Mass Rapid Transit (MRT) trains form the backbone of the railway system. There are also tens of thousands of taxicabs operated by commercial companies and by individual taxi owners [10]. IC cards are widely used for paying transit fees. Several card readers are deployed for collecting the fees on SBS and SMRT public buses and at entrance gates of MRT stations.

We experiment in both campus shuttle buses and public transport buses (SBS Transit bus service in Singapore). As shown in Figure 16, there are 4 shuttle bus routes (i.e., Route A-D) in our campus. The shuttle buses serve from 08:00 to 23:00 with time intervals varying from 5 to 20 minutes. The bus route lengths span approximately from 3.8km to 5.8km with celltower set sequence lengths varying from 9 to 22. The average velocity of the buses is about 20km/h. Table 3 gives the details of the bus routes. The shuttle bus routes have overlapped road segments as depicted in Figure 16. The campus bus C travels in clockwise direction, while buses A, B, and D move in counterclockwise direction. We see that Route A and Route C have substantial overlapped segments. Table 4 summarizes the shared route segments between each pair of bus routes, which span from 0km to 3.4km. We see that around 85% (3.4km/4km) of Route A overlaps with Route C.

We experiment on SBS Transit bus route 179 and 241 as well. For comparison study, we also collect celltower sequences and accelerometer readings in East-West and the North-South MRT Lines in Singapore.

4.2 Bus detection performance

4.2.1 Audio detection accuracy

We collect more than 200 beep signals on different *public transit buses* during our 7-week experiments. We set the audio sampling rate to be 8kHz, and we use 128-pt FFT to detect the IC card reader. We test the bus detection method by varying the distances between the IC card reader and the mobile phones (approximately 1 meter to 7 meters). We also consider the scenarios where mobile phones may be held in hand and inside bags. We report the average detection accuracy of single beeps in different circumstances. In Figure 17, we see that the detection rate is over 95% when mobile phones are in close vicinity to the IC card reader (e.g., within 3 meters) even when they are placed in bags. With mobile phones placed 5 meters away from the reader, the detection accuracies are about 58% held in hand, and 71% placed in bags, respectively. As the distance increases further (e.g, >7 meters), the detection accuracy drops substantially. In addition, we list the detection rate, false positive rate, and accuracy of bus detection method in Table 5.

The experiment results suggest that the audio based method effectively detects the beep signal on the bus when the distance between the IC card reader and the mobile phone is within 3 meters. Considering that the entrance gate of the bus is about 1.4 meters wide, when a sharing user enters a bus, the mobile phone would be less than 1 meter away from the IC card reader (normally within 0.5 meters).

4.2.2 Bus vs. MRT train

We next evaluate the accelerometer based bus detection method that is used to distinguish the buses from the MRT trains. Figure 18 plots the accuracy in detecting the buses. We find that accelerometer based method can distinguish the buses from the MRT trains with an accuracy of over 90% on average. We analyzed the main reason for falsely detecting public buses as MRT trains, and find that it happens mostly when the buses are driving along long straight routes late during night time. The accelerometer readings may be relatively stable and very similar to those on the MRT trains.

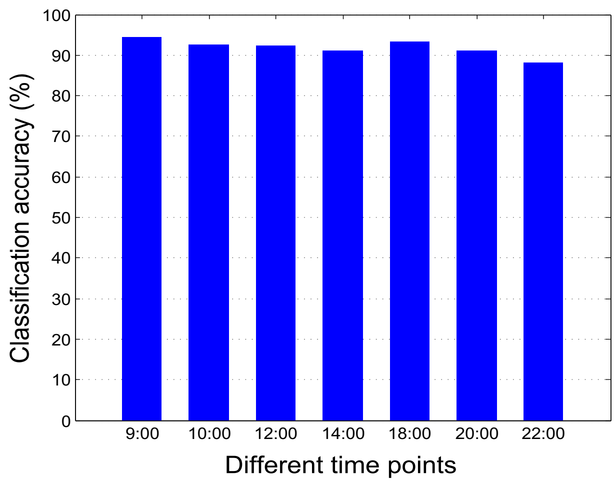


Figure 18: Bus vs. MRT using accelerometer

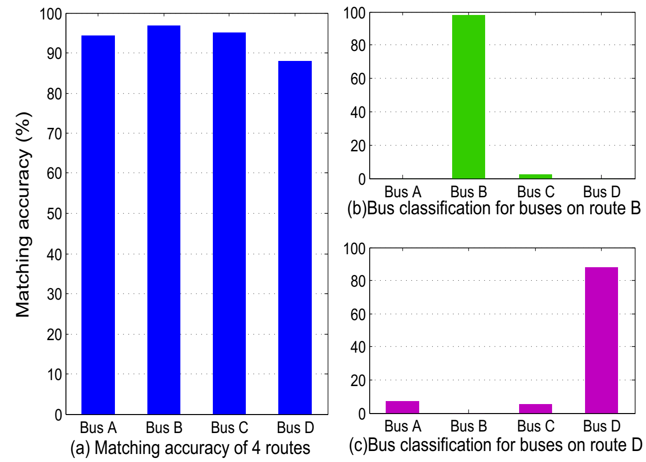


Figure 19: Bus classification accuracy

4.3 Bus classification performance

We present the evaluation results for our bus classification algorithms. In our prototype system, we collect the celltower sequences on the 4 campus bus routes and store them in the database. The campus buses do not have IC card readers, so we use the GNUradio to produce and play the dual-tone (1kHz and 3kHz) beeps. Mobile phones start to collect data after detecting the beeping signals on buses. For the public transit buses (e.g., SBS transit and SMRT Corporation buses), the mobile phones can directly detect their IC card readers. The data collection process spans over a period of 7 weeks. We collect 20 runs for each shuttle bus route for the bus route classification. As the cellular networks are likely to be updated incrementally, most celltowers along the bus routes typically remain consistent during the experiment period.

We implement the celltower sequence matching with the top-*3* celltower sequence matching algorithm. In Figure 19(a), we plot the bus classification results for the 4 campus bus routes. According to the experiment results, the bus classification accuracy is approximately 90% with the highest accuracy of 96% for Bus B and the lowest of 87% for Bus D. Although 85% of Route A is overlapped with Route C, the bus classification accuracy for Bus A and C are still around 94%. The main reason is that Bus A and C travel in the opposite directions. Since Route D shares a large portion of overlapped road segments with Route A and Route C, and buses travel in the same direction on the shared road segments, buses along Route D might be misclassified to Route A or Route C. Figure 19(c) depicts the classification ratio of buses along Route D. We can find that 7% of the buses are misclassified to Route A and 6% are misclassified to Route C. Although Route B has many overlapped road segments with Route A and C, the buses travel in the opposite directions on those road segments. (Figure 19(b)) depicts the classification ratio of buses along Route B. We find that only 3% of the buses are misclassified to Route C. Overall, the bus classification accuracy is satisfactory, considering the high overlap ratio of the four routes in the campus (the city-wide public bus routes are far less overlapped, e.g., SBS 179 and 241).

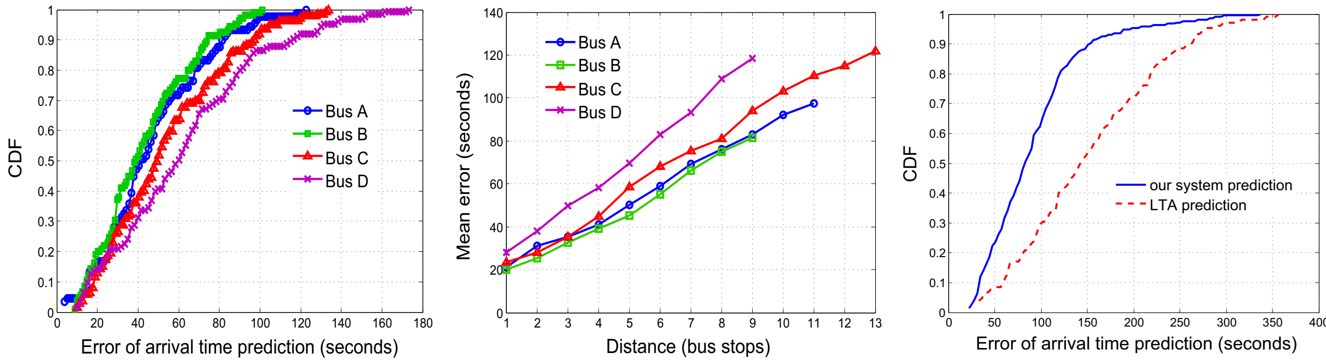
4.4 Arrival time prediction

We present the final bus arrival time prediction results based on above estimations. We collect the campus bus traces using a high accurate vehicle GPS navigator as the benchmarks. In the same buses, we collect celltower sequences using two mobile phones and stored the sequence in memory stick for our later trace-driven study.

In the trace-driven study, we generate queries at different campus bus stops according to poisson arrival process, and compare the predicted arrival time with the actual arrival time of the campus buses to compute the average of the absolute prediction error. Figure 20(a) shows the CDF of the absolute error of arrival time prediction results. The median prediction errors vary approximately from 40s for Bus B to 60s for Bus D. The 90th percentiles are approximately from 75s for Bus B to 115s for Bus D, respectively. Generally, the average estimation error increases as the length of bus route increases.

Figure 20(b) plots the average error against the distance between the sharing user and the querying user, where we approximate the distance using the number of bus stops. We observe that as the bus moves closer to the querying user, the prediction error becomes smaller. The error of Bus D increases faster than those of Bus A, B, and C.

We experiment with commercial bus system as well. For comparison, we also query the arrival time of public transit buses provided by LTA of Singapore. The public buses are periodically tracked with on-bus localization devices and respond to the queries for the bus information. People can send an SMS to query the bus arrival time indicating the interested bus route and stop. In the experiment we test the arrival time prediction on SBS bus route 179 and 241. We compute the prediction error by comparing the predicted results with the actual arrival time of the buses. Both prediction errors of LTA and our system are measured and we plot the CDF of the prediction results in Figure 20(c). According to the results, the average prediction error of our system is approximately 80 seconds, while the prediction result of LTA is around 150 seconds. Such a comparison result is surprising, as we expect more accurate prediction result from the commercial system of LTA where a rich set of resources including on-bus GPS sensors are proactively used.



(a) Bus arrival time prediction error (b) Bus arrival time prediction (c) Our system v.s. LTA

Figure 20: Arrival time prediction performance

Sensors	Samsung i9100	HTC Desire
No sensor	18.2	15.3
Accelerometer 20Hz	18.0	15.2
Microphone 8kHz+FFT	17.5	14.9
Celltower 1Hz	17.8	15.0
GPS 1Hz	9.7	6.4

Table 6: Battery duration for different sensor settings (in hours)

We suspect that the deployed system of LTA is intentionally made inaccurate (e.g., using caching to reduce computation and communication cost), yet we cannot further dig into such a commercially running system for more details.

4.5 System overhead

Mobile phone. The computation complexity of the algorithms on mobile phones is bounded by the length of audio signals and accelerometer signals needed for the bus detection. In order to maintain the sample resolution and remove the noise, we extract the audio signal with sliding widows with the window size of 32. We record the audio signal at the sampling rate of 8kHz, and use $n = 128$pt FFT to convert the time domain audio signals to frequency domain signals. The major computational complexity is attributed to performing FFT on mobile phones which is $O(n \log n)$. Current mobile phones can finish the computation task in realtime. For example, it takes approximately 1.25ms and 1.8ms on average to finish to 128pt FFT on Samsung Galaxy S2 i9100 and HTC Desire, respectively.

We measure the power consumption of continuously sampling microphone, accelerometer, GPS, and cellular signals. Table 6 illustrates the measured battery lifetime when the mobile phones continuously trigger different sensors. The experiments were performed with the screen set to minimum brightness. We report the average results over 10 independent measurements. The battery duration was quite similar for sampling accelerometer at 20Hz, sampling audio signal at 8kHz with 128pt FFT, and sampling no sensors. Sampling the celltower signal consumes limited extra battery power as well. On the other hand the battery lifetime is substantially reduced when the GPS module in the phone is enabled.

Backend server. The computation overhead of backend server is mainly bounded by the bus classification algorithm, i.e., the uploaded celltower sequence length l, the celltower set sequence length k, and the number of celltower set sequences in the database N. The computation complexity of sequence matching using dynamic programming is $O(lk)$, and as we need to compare with N candidate sequences in database the overall computation complexity is $O(lkN)$. Since in practice both m and n are usually small (e.g., max$\{l,k\}$ is around 40 according to our experiments), the computation complexity increases almost linearly to the number of candidate celltower sequences in the database.

5. LIMITATIONS AND ON-GOING WORK

Alternative referencing points. In practical implementation, we observe that the number of celltowers that a sharing user can capture on a bus influences the bus classification accuracy. It takes a few minutes for the passenger on a bus to observe several celltowers to form a reliable sequence for bus classifications. We are currently studying to utilize the ambient radio signals and extract useful information (e.g., WiFi points) to complement the celltower IDs as fingerprints. As a matter of fact, there are relatively stable WiFi points along different bus routes. Similar to using GPS modules, such a method consumes extra power on the mobile phones, though the WiFi module draws much less power. Our preliminary measurement indeed finds many reliable WiFi hotspots in our campus which may extend the celltower fingerprints. We are also studying how such complementary information can be utilized in an energy efficient manner.

Number of passengers. The number of sharing passengers affects the prediction accuracy in our system. When there is no sharing passengers on a bus, the backend server would miss the bus, which affects the prediction results. This common issue of crowd-sourced solutions is largely influenced by the penetration rate and popularity of the services. One may actively promote the service to reach a critical penetration rate so as to ensure that at least one sharing user is on the bus willing to report the bus status. At the initial stage, we may equip the bus driver with the mobile phone clients so that at least one sharing user (i.e., the bus driver) can update the bus status to the backend server.

First few bus stops. The bus classification method needs a sufficiently long celltower sequences for accurate bus route classification, and consequently the arrival time at the first few bus stops would not be timely updated. As discussed in [12], being less affected by unpredictable traffic conditions, the arrival time at the first few bus stops is stable and predictable according to the historical data. Therefore, the backend server may bias the arrival time prediction for those bus stops towards the official bus schedules and real-time traffic conditions on the bus routes.

Overlapped routes. By concatenating several celltower sequences from the same bus, the backend server may obtain a longer celltower sequence for bus classification. Although such a longer sequence alleviates the overlapped bus route issue, occasionally our bus classification algorithm still cannot confidently classify the bus routes and have to postpone responses to querying passengers. Since many buses typically arrive at and depart from several major transit centers, such route overlapping could be especially common in downtown area. In our experiments, we find that there are distinct differences of the bus speeds between the campus shuttle buses and the public transit buses even when the buses travel the same overlapped sub-routes. The buses with more frequent bus stops tend to drive at lower speed than those with fewer bus stops. We are currently exploring how to effectively utilize the differences of the bus speed to distinguish the buses with overlapped routes. The problem of efficiently and accurately classifying different bus routes sharing substantial portion of overlapped segments (e.g., in downtown areas) remains challenging.

6. RELATED WORK

Phone-based transit tracking. Our work is mostly related to recent works on the transit tracking systems [12, 20, 29]. EasyTracker [12] presents an automatic system for low-cost, real-time transit tracking, mapping and arrival time prediction using GPS traces collected by in-vehicle smartphones. Thiagarajan *et al.* [29] present a grassroots solution for transit tracking utilizing accelerometer data and GPS modules on participating mobile phones. Our work differs from them in that it predicts the bus arrival time based on celltower sequence information shared by participatory users. To encourage more participants, no explicit location services (e.g., GPS-based localization) are invoked so as to reduce the overhead of using such special hardware for localization.

EEMSS [32] presents an energy efficient sensor management framework which uses minimum number of sensors on mobile devices to monitor user states. Nericell [23] uses onboard sensors to efficiently monitor the road surface quality and traffic conditions. VTrack [31] predicts road traffic time based on a sequence of WiFi-based positioning samples using an HMM-based algorithm for map matching. Ravindranath *et al.* [27] use various sensor hints to improve wireless protocols. CTrack [30] presents energy-efficient trajectory mapping using celltower fingerprints and utilizes various sensors on mobile phones to improve the mapping accuracy. Balan *et al.* [10] present a realtime trip information system to predict taxi fares and trip time. SignalGuru [19] presents a software service that predicts traffic signals' future schedule which enables green light optimal speed advisory by leveraging opportunistic sensing on windshield-mount smartphones. Yang *et al.* [35] present a driver detection system that distinguishes a driver and a passenger leveraging car speakers and mobile phone microphones.

Celltower sequence matching. StarTrack [7] provides a comprehensive set of APIs for mobile application development. Applying new data structures, [17] enhances StarTrack in efficiency, robustness, scalability, and ease of use. CAPS [25] determines a highly mobile user's position using a cell-ID sequences matching technique which reduces GPS usages and saves energy on mobile phones. Unlike those proposals, our work does not aim to position the mobile users though similar in spirit to these existing works in utilizing the celltower sequences.

Participatory sensing. Many recent works develop participatory platforms for people-centric mobile computing applications [6, 13]. Micro-blog [16] presents a participatory sensing application which connects sharing parties and querying parties to allow geo-tagged multimedia sharing. MoVi [11] studies the problem of social activity coverage where participants collaboratively sense ambience and capture social moments through mobile phones. SoundSense [22] classifies ambient sounds to achieve context recognition. SurroundSence [8] utilizes various sensors on mobile phones to collect identifiable fingerprints signals for logical localization. Escort [14] obtains cues from social encounters and leverages an audio beacon infrastructure to guide a user to a desired person. WILL [34] designs an indoor logical localization technique leveraging user mobility and WiFi infrastructure while avoiding site survey. Although targeted at totally different applications and problems, the common rationale behind these works and our design is that the absolute physical location of users though sometimes sufficient not always necessary to accomplish particular tasks.

7. CONCLUSIONS

In this paper, we present a crowd-participated bus arrival time prediction system using commodity mobile phones. Our system efficiently utilizes lightweight onboard sensors which encourages and attracts participatory users. Primarily relying on inexpensive and widely available cellular signals, the proposed system provides cost-efficient solutions to the problem. We comprehensively evaluate the system through a prototype system deployed on the Android platform with two types of mobile phones. Over a 7-week experiment period, the evaluation results demonstrate that our system can accurately predict the bus arrival time. Being independent of any support from transit agencies and location services, the proposed scheme provides a flexible framework for participatory contribution of the community.

8. ACKNOWLEDGEMENT

We would like to thank our shepherd, Rajesh Krishna Balan, as well as the anonymous reviewers for providing constructive feedbacks and valuable input for improving the quality of this paper. We acknowledge the support from NTU SUG M4080103.020 and NAP M4080738.020.

9. REFERENCES

[1] Bus transport in Singapore. `http://en.wikipedia.org/wiki/Bus_transport_in_Singapore`.

[2] EZ-Link. `http://www.ezlink.com.sg`.

[3] Octupus. `http://www.octopus.com.hk/home/en`.

[4] Oyster. `https://oyster.tfl.gov.uk/oyster`.

[5] PublicTransport@SG. `http://www.publictransport.sg/`.

[6] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for Human Spaces. *IEEE Pervasive Computing*, vol. 6(issue 2): pages 20–29, Apr. 2007.

[7] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath. Startrack: a framework for enabling track-based applications. In Proceedings of ACM MobiSys, pages 207–220, 2009.

[8] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In Proceedings of ACM MobiCom, pages 261–272, 2009.

[9] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In Proceedings of IEEE INFOCOM, pages 775–784, 2000.

[10] R. K. Balan, K. X. Nguyen, and L. Jiang. Real-time trip information service for a large taxi fleet. In Proceedings of ACM MobiSys, pages 99–112, 2011.

[11] X. Bao and R. Roy Choudhury. Movi: mobile phone based video highlights via collaborative sensing. In Proceedings of ACM MobiSys, pages 357–370, 2010.

[12] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson. Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In Proceedings of ACM SenSys, pages 1–14, 2011.

[13] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In Workshop on World-Sensor-Web (WSW): Mobile Device Centric Sensor Networks and Applications, pages 117–134, 2006.

[14] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury. Did you see bob?: human localization using mobile phones. In Proceedings of ACM MobiCom, pages 149–160, 2010.

[15] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In Proceedings of ACM MobiSys, pages 49–62, 2010.

[16] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: sharing and querying content through mobile phones and social participation. In Proceedings of ACM MobiSys, pages 174–186, 2008.

[17] M. Haridasan, I. Mohomed, D. Terry, C. A. Thekkath, and L. Zhang. Startrack next generation: a scalable infrastructure for track-based applications. In Proceedings of USENIX OSDI, 2010.

[18] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles. Pbn: towards practical activity recognition using smartphone-based body sensor networks. In Proceedings of ACM SenSys, pages 246–259, 2011.

[19] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In Proceedings of ACM MobiSys, pages 127–140, 2011.

[20] F. Li, Y. Yu, H. Lin, and W. Min. Public bus arrival time prediction based on traffic information management system. In Proceedings of IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pages 336–341, 2011.

[21] Y. Liu, L. Chen, J. Pei, Q. Chen, and Y. Zhao. Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. In Proceedings of IEEE PerCom, 2007.

[22] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In Proceedings of ACM MobiSys, pages 165–178, 2009.

[23] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In Proceedings of ACM SenSys, pages 323–336, 2008.

[24] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In Proceedings of ACM MobiSys, pages 299–314, 2010.

[25] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In Proceedings of ACM MobiSys, pages 293–306, 2011.

[26] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In Proceedings of ACM SenSys, pages 1–14, 2007.

[27] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden. Improving wireless network performance using sensor hints. In Proceedings of USENIX NSDI, 2011.

[28] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks*, vol. 6(issue 2): pages 1–27, March 2010.

[29] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In Proceedings of ACM SenSys, pages 85–98, 2010.

[30] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod. Accurate, low-energy trajectory mapping for mobile devices. In Proceedings of USENIX NSDI, 2011.

[31] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In Proceedings of ACM SenSys, pages 85–98, 2009.

[32] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In Proceedings of ACM MobiSys, pages 179–192, 2009.

[33] M. S. Waterman and T. F. Smith. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[34] C. Wu, Z. Yang, Y. Liu, and W. Xi. WILL: Wireless indoor localization without site survey. In Proceedings of IEEE INFOCOM, 2012.

[35] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin. Detecting driver phone use leveraging car speakers. In Proceedings of ACM MobiCom, pages 97–108, 2011.