

Tracking Human Queues Using Single-Point Signal Monitoring

Yan Wang[†], Jie Yang[‡], Yingying Chen[†]
Hongbo Liu[‡], Marco Gruteser^{*}, Richard P. Martin^{*}

[†]Stevens Institute of Technology, Hoboken, NJ 07030, USA

[†]{[ywang48](mailto:ywang48@stevens.edu), [yingying.chen](mailto:yingying.chen@stevens.edu)}@stevens.edu

[‡]Oakland University, Rochester, MI 48309, USA

[‡]yang@oakland.edu

[‡]Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, USA

[‡]hl45@iupui.edu

^{*}Rutgers University, North Brunswick, NJ 08902, USA

^{*}gruteser@winlab.rutgers.edu, rmartin@cs.rutgers.edu

ABSTRACT

We investigate using smartphone WiFi signals to track human queues, which are common in many business areas such as retail stores, airports, and theme parks. Real-time monitoring of such queues would enable a wealth of new applications, such as bottleneck analysis, shift assignments, and dynamic workflow scheduling. We take a minimum infrastructure approach and thus utilize a single monitor placed close to the service area along with transmitting phones. Our strategy extracts unique features embedded in signal traces to infer the critical time points when a person reaches the head of the queue and finishes service, and from these inferences we derive a person's waiting and service times. We develop two approaches in our system, one is directly feature-driven and the second uses a simple Bayesian network. Extensive experiments conducted both in the laboratory as well as in two public facilities demonstrate that our system is robust to real-world environments. We show that in spite of noisy signal readings, our methods can measure service and waiting times to within a 10 second resolution.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

Keywords

Human Queue Monitoring; WiFi; Smartphones; Received Signal Strength

1. INTRODUCTION

The popular, almost addictive, usage of smartphones and their data-intensive apps creates novel opportunities to exploit their network traffic for monitoring and optimizing real-world processes. Research has shown, for example, how cellular call data records

can be used to infer large scale transportation patterns (e.g., [13]) or how cellular signal traces allow inferring the level of congestion on roadways [4]. In this work, we ask whether signal power readings from cell phone traffic are also sufficient to monitor a much finer-scale, yet common, process: human queues.

Such queues are a familiar and often frustrating occurrence, for example in retail stores, banks, theme parks, hospitals and transportation stations. Figure 1 shows the abstraction we map onto these environments. As people arrive, the *waiting period* is the time spent waiting for service. During the *service period* people receive service, such as paying for items or checking-in travel bags. A person exits the service area during the *leaving period*. Note that we interpret the concept of a queue loosely, people do not need to stand in line but could sit in a waiting room and do not always need to be served in a strict first in, first out order. Real-time quantification of the waiting and service times in such queues allows optimizing service processes, ranging from retail, to health care, to transportation and entertainment. For example, many hospital emergency departments surveys have average waiting times of several hours [21]. More complete waiting and service time statistics allow customers, travelers, managers and service providers make changes to their behavior and processes. For example, an airport checkpoint might be experiencing abnormal delays and require interventions by diverting screeners from queues with shorter waiting times. Customers also can benefit, for example, knowing at what times retail store checkout lines can be expected to be shorter, and a customer can decide whether to stay in the queue or go to do more urgent tasks. Managers can use such information to make staffing decisions based on the service length. For example, during particular hours in a day, service times may grow at a coffee shop due to increased demands for espresso drinks compared to other items. In such a case, it might be more effective to change the staffing to use experienced baristas as opposed to simply adding staff. A hospital emergency department may shift nursing staff to assist with triage when waiting times become too long. In the transportation field, bus and train schedules or boarding and payment processes could be adjusted.

Existing solutions to the queue monitoring problem rely on cameras [1, 16] or special sensors (e.g., infrared [24] or floor mats [2]) and usually require sensors at multiple locations. The approaches using cameras face occlusion and increased privacy issues. Bluetooth signals emanating from phones have also been used to measure travel times between two sensing points, both at airports [3]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
MobiSys'14, June 16–19, 2014, Bretton Woods, New Hampshire, USA.
Copyright 2014 ACM 978-1-4503-2793-0/14/06 ...\$15.00.
<http://dx.doi.org/10.1145/2594368.2594382>.

and for vehicle traffic [23]. These prior techniques using wireless networks were too coarse-grained to differentiate between the waiting and service time. Moreover, all these solutions require multiple sensors to fully monitor a single longer queue, which increases installation and system cost.

Our approach uses only a single sensor, a WiFi monitor near the head of the queue that measures the received signal strength of packets emitted from phones. Intuitively, the received signal power should follow a known pattern. As the person moves towards the service point, the phone moves closer to the monitor and the received signal power should increase. When the person is receiving service, the signal power should be very strong and relatively constant. Finally, when the person exits the service point, there should be a rapid decline in signal strength. Our approach seeks to exploit this pattern to track service and waiting times of persons in a queue. This does require, of course, that at least some people in the queue carry phones that generate traffic. In our existing system and experiments, we have used a special app to generate periodical beacons. In the future, we expect, however, that it is possible to reuse existing traffic, since it is increasingly the case as people tend to fiddle with their phones while waiting in queues and many venues now offer loyalty apps that customers may use while paying, for example, the Starbucks mobile app has more than 10 million active users [11]. While this approach may also be applicable to other wireless technologies, we have chosen WiFi in our realization because (i) its range is sufficiently large to cover the entire queue in most cases, especially for large-size queues, which usually requires the installation of multiple cameras or sensors in order to obtain the complete view of the queue; (ii) it is particularly easy to monitor WiFi traffic (compared to the complexities tracking Bluetooth frequency hopping sequences, for example); (iii) WiFi traffic is poised to increase due to the WiFi offloading trend [7].

Accurately discerning the points where the person begins and ends service is challenging, however, in this single-point monitoring system, because the multi-path, shadowing, and fading components of a signal are quite dynamic due to the movement of many people. We therefore investigate two approaches for extracting the waiting and service times from a signal trace: a direct feature-driven one and a Bayesian Network one. For both approaches, we use similar filtering techniques to reduce the impact of the dynamic environment. We experimentally evaluate both approaches in three settings: a laboratory, a coffee shop and an airport. We find a simple Bayesian classifier gives the best results, and is able to measure the waiting time to within average errors less than 10s and maximum errors below 20s. Our contributions are:

- Proposing a single-point WiFi monitoring approach for human queue measurement systems that requires less infrastructure and can provide real-time service information for both business and customers.
- Designing queue measurement techniques that can distinguish between important time periods in human queues including waiting, service, and leaving times using unique WiFi signal patterns based on feature extraction and Bayesian networks.
- Leveraging the availability of multiple antennas and multiple smartphone users in the queue to accurately perform queue parameter estimation.
- Demonstrating through extensive experiments with participants randomly placed in a variety of queue patterns in real environments (e.g., a coffee shop and an airport) that it is

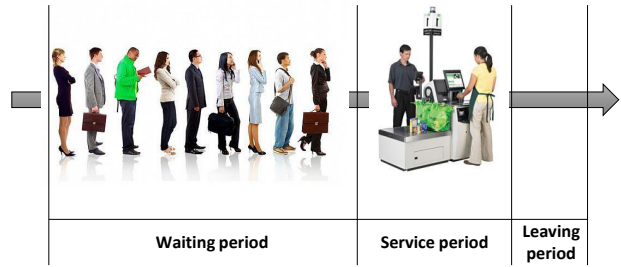


Figure 1: Important time periods and corresponding positions in a human queue.

feasible to track human queues using single-point WiFi monitoring with high accuracy.

The rest of the paper is organized as follows. In Section 2, we investigate whether a sufficient amount of WiFi users are present in human queues to facilitate queue measurements. We describe the challenges, system overview, and two core schemes to measure parameters of human queues in Section 3. We present our system implementation by leveraging multiple antennas and smartphone user traces in Section 4. In Section 5, we perform extensive evaluation of our system in real environments, i.e., a coffee shop and an airport. We discuss limitations in Section 6 and put our work into the context of related research in Section 7. Finally, we conclude in Section 8.

2. PREVALENCE OF WIFI TRAFFIC IN QUEUES

In this section, we investigate whether a sufficient number of WiFi users are present in human queues to facilitate queue measurements.

2.1 WiFi Traffic in Human Queues

An increasing number of facilities, such as retail stores, shopping malls, or airports, offer Internet access through WiFi networks to provide convenience to customers. People visiting these places tend to use the Internet through the WiFi connection because it provides faster speeds than the 3G network [10, 17] and does not count against their data cap. In addition, the increasing trend of the WiFi offloading enables smartphones to automatically switch to WiFi in popular locations where the cellular carriers operate WiFi hotspots [7]. This can be expected to lead to increasing WiFi usage in these locations.

We start by investigating the WiFi device density in a coffee shop over a one month time period. Our objective is to find the percentage of people generating WiFi traffic with their smartphones while standing in the line waiting for service. We utilize a WiFi monitor, placed close to the service desk, to passively monitor WiFi packets sent by mobile devices. To eliminate WiFi devices not in the queue, we use the heuristic that devices with a higher received signal strength (RSS) amplitude, i.e. greater than $-45dBm$, are likely to be in the queue. We have empirically determined this threshold by observing the RSS of a known smartphone placed in the queue close to the service desk. We use the MAC address extracted from captured packets to identify different devices. Meanwhile, we manually count the arrival time of each new customer and the length of the queue as the ground-truth for comparison.

Date	1/21 (noon)	1/28 (noon)	2/3 (noon)	2/7 (noon)	2/12 (noon)	2/14 (morn.)	2/16 (noon)
Time length (hrs)	2	2	2	1.5	1	1	2.5
Discoverable WiFi devices in the queue (w. Max. RSS > -45dBm)	24	36	88	26	37	45	84
Discoverable Bluetooth devices in the queue	7	7	6	7	6	6	8
People in the queue (manually counted)	153	82	205	78	88	87	128
Average length of the queue	2.55	1.83	5.19	4.03	3.55	5.47	5.38
Discoverable WiFi devices in the store	307	280	410	278	364	384	608

Table 1: Data collected in a coffee shop from seven days time period.

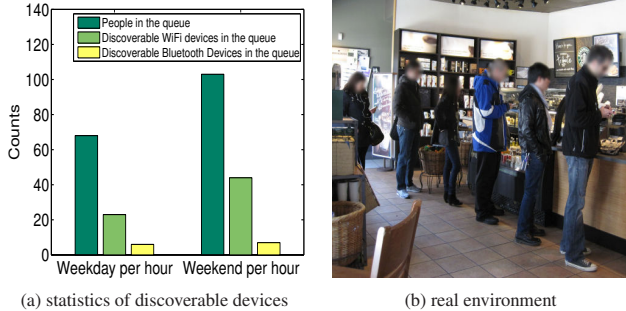


Figure 2: Comparison of the manually counted number of people, discoverable WiFi and Bluetooth devices in queues within a coffee shop, and the illustration of customers using smartphones while waiting for service.

We show detailed data collected for seven days in Table 1. An interesting observation is that the number of discoverable WiFi devices in queues is higher when the average queue length is longer. For example, on both Feb. 3rd and Feb. 16th of 2013, the discoverable WiFi devices in queues are almost doubled compared to other days. This observation implies that a longer waiting time triggers people in the queue to use smartphones more, thus generating more WiFi traffic. We present the comparison of discoverable WiFi devices with a manual count of people in the queue over the one-month study in Figure 2. It shows that statistically there are about 34% customers using WiFi (most likely using smartphones) while waiting in queues in the coffee shop during weekdays and increases to 43% during weekends. While this analysis is based on heuristics, the result is encouraging as it indicates that the number of WiFi signal emitting devices in the queue may already be large and can be exploited for real-time human queue measurements.

2.2 Bluetooth Signals in Human Queues

Since Bluetooth is another wireless technology that has been widely used in mobile applications, we also examine the Bluetooth signals presented in a human queue of the same coffee shop for comparison. We place a Bluetooth dongle close to the service desk running a self-developed software program under Ubuntu 10.04, which can detect Bluetooth devices that are in *discoverable mode*. Because most Bluetooth modules used in mobile devices have a communication range of about 10 meters [17], all devices captured by the Bluetooth dongle are considered valid devices close enough to be within the queue. We find many few Bluetooth devices than WiFi devices in the same store on the same days. Figure 2 shows that there are about 74% and 84% less discoverable Bluetooth devices than WiFi devices during weekdays and weekends respectively in the queue. In principle, our technique should also work with Bluetooth signals, particularly with the expanded range of newer Bluetooth versions. These results support, however,

that WiFi may already be an attractive alternative since monitoring Bluetooth communications requires more effort.

3. SYSTEM DESIGN

In this section, we discuss the challenges and goals of our approach and provide an overview of our system.

3.1 Challenges and Goals

We explore the approach of using low infrastructure and low cost sensing technology that can infer important queueing times from smartphone WiFi signals. Our goal is to employ a single monitor and utilize the smartphones carried by the people waiting in the queue to perform real-time tracking of human queues. To implement such a system, we need to address a number of challenges.

Tracking Queues Using a Single Monitor. If multiple monitors (e.g., WiFi hotspots) were employed, the phone could be localized and hence the queue parameters could be derived from the localization results. However, installing additional monitors are quite costly due to installation costs. The low infrastructure approach by using only a single WiFi monitor cannot uniquely determine the phone’s position. To address this challenge, our solution should be able to identify the unique characteristics presented in the smartphone’s Received Signal Strength (RSS) traces, which may be utilized to perform queue parameter estimation without the need of explicitly localizing the phone.

Robustness to Real Environments. Although the distance between the smartphone and the service desk dominates the RSS, the RSS is affected by a number of factors including user movements, changing environments, and signal interference. Furthermore, different holding styles and vibrations of smartphones also cause noisy RSS readings. Thus, the system should be designed in such a way that can cope with noisy signal readings.

Identifying Queue Related RSS Traces. The RSS traces extracted from smartphones cannot be directly used to estimate queue parameters, since there could be trends of the received signal happening after the queue process that are similar to the RSS trend within the queue. An effective data calibration mechanism is required to identify the right segment of the RSS trace that includes only the important periods of queue process, i.e., the waiting period, service period, and leaving period.

3.2 System Overview

By placing a WiFi monitor close to the service area, our system can monitor human queues in real-time through examining the unique patterns exhibited in the WiFi signals, which are extracted from smartphones carried by people waiting in the queue. It can also utilize a WiFi hotspot if it is located close to the service area. Our system exploits the change of RSS trends from smartphones carried by people going through a queue process. We find that with the monitor located at the service area, each phase of the queue creates unique signal patterns that can be used to identify different

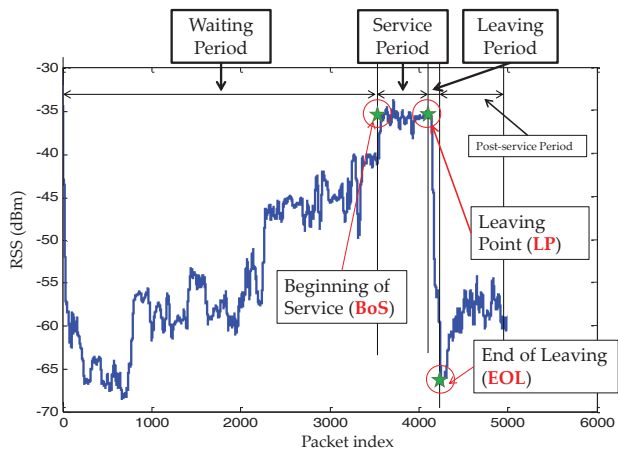


Figure 3: Illustration of special queue-related patterns embedded in the RSS trace collected from a smartphone in a human queue.

periods of the queue for statistical queue measurement. For example, Figure 3 presents the RSS trace of a smartphone in a queue collected from a single WiFi monitor at the service desk in a coffee shop. We observe that the captured RSS trace reflects the pattern of the distance between the person carrying the phone to the service desk. In particular, the RSS exhibits the following unique patterns: (1) the RSS has a slowly increasing trend during the *waiting period*; (2) it then becomes stable at high RSS values when the person carrying the phone is in the *service period*; and (3) it finally drops quickly when the person leaves the service desk after having received service in his *leaving period*.

The major advantage of this solution is that it can work under real-world queue scenarios in variant environments without requiring a specialized infrastructure and incurring manpower overheads. In addition, our solution is based on the detection of the important time points associated with the service period, which involves the strongest and the most stable signal strength of the smartphone when it is at the closest position to the service desk (WiFi monitor), thus it can work in typical indoor environments with multi-path effects. Furthermore, our solution does not require all the people in the queue to carry phones. Instead, it only requires a sample of phones generating WiFi traffic to track human queues. This is increasingly the case: since people tend to use their phones in WiFi Hotspots when waiting in a queue, and many venues provide customer loyalty apps for people to use in the stores, and these could also be customized. Our feasibility analysis in Section 2 confirms this observation.

We realize our approach with three main sub-tasks: *Data Calibration*, *Integration of Multiple Antennas*, and *Queue Parameter Determination*. Figure 4 shows the flow of our system. We assume the user gets in the queue right after he enters the facility (e.g., a customer enters a coffee shop to buy coffee or a passenger gets on a platform waiting for a train). When the smartphone carried by the user is discovered by the WiFi monitor as the user goes in the queue, the WiFi monitor extracts RSS readings of this device. Based on the RSS trace, our system first performs *Data Calibration*, which aims to preserve the unique trend presented in the raw RSS trace while removing high frequency noise, and to identify segments of the RSS trace that contains related time periods for measuring queue parameters. The *Data Calibration* sub-task contains three steps: *Interpolation*, *Low-pass Filtering* and *Coarse Queue Identification*.

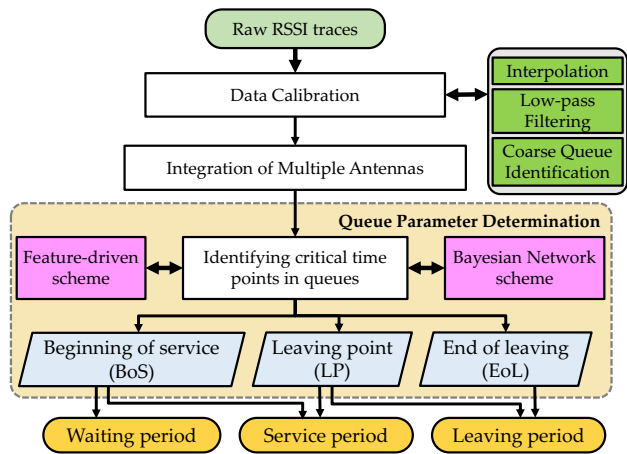


Figure 4: Overview of system flow.

To filter out outliers and obtain a reliable RSS trace, the system further conducts *Integration of Multiple Antennas* by exploiting the availability of multiple antennas, which exists in many WiFi access points already. This subtask combines the selected RSS traces from two antennas in the WiFi monitor to generate an integrated RSS trace that fortifies the unique pattern of RSS associated with important time periods of the queue. Finally, *Queue Parameter Determination* implements two schemes, called: feature-driven and Bayesian Network, to infer the critical time points in the queue. The important time points: the beginning of service (BoS), leaving point (LP) and end of leaving point (EoL) are further utilized to estimate the queue parameters including waiting period, service period, and leaving period. We next present the two schemes developed in the core component *Queue Parameter Determination*. We leave the detailed presentation of *Data Calibration* and *Integration of Multiple Antennas* to the next section (Section 4).

3.3 Queue Parameter Determination

We find that the important periods of a human queue are separated by three critical time points, namely BoS, LP, and EoL, as shown in Figure 3. BoS is the first critical time point, which separates the waiting period and the service period, while LP tells when the service is done and the customer is about to leave the service desk. The third critical time point, EoL, indicates the time that the customer has left the service area, such as goes to a seat or exits the facility. By deducing these three time points, our system is able to track the important queue time periods including waiting, service, and leaving periods. Specifically, the *waiting period* is the time interval between BoS and the starting time of the trace, whereas the *service period* is the time interval between BoS and LP. And the *leaving period* is the time interval between LP and EoL.

Toward this end, we develop two schemes, called *feature-driven* and *Bayesian Network*, in our system to identify the critical time points in a human queue. The feature-driven approach utilizes the unique features extracted from RSS traces to identify these time points. Whereas the Bayesian Network scheme models the detection of BoS and LP as a probabilistic graph based on the statistics of features presented in RSS traces. Comparing these two approaches, the feature-driven scheme requires less prior knowledge on the queue while the Bayesian Network scheme produces more accurate results bounded by the probabilistic graph. We next describe the details of these two approaches.

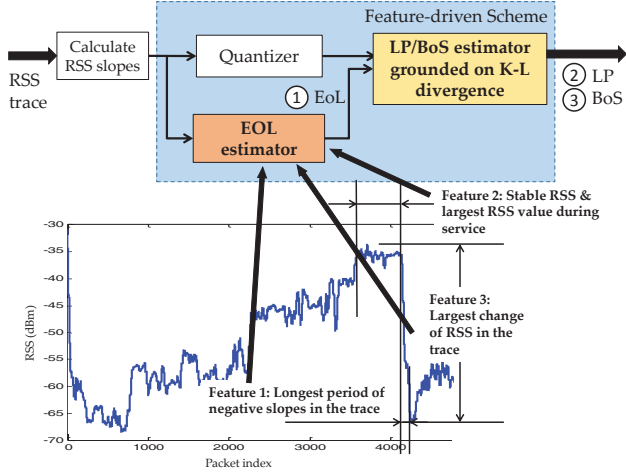


Figure 5: Flow of feature-driven scheme to find EoL, LP, and BoS in a time-reversed way.

3.3.1 Feature-Driven Scheme

When designing the feature-driven scheme, we directly apply features associated with the leaving period in the RSS trace to determine the EoL, LP, and BoS in a time-reversed manner. Because the RSS value changes dramatically when people leave the queue after service, the features associated with the leaving period are easier to extract in a reverse order. In particular, we identify three features extracted from the RSS trace associated with the leaving period:

1. The leaving period has the *longest* consecutive negative-slope segments of the selected RSS trace.
2. The RSS values before the leaving period are stable with the *highest* amplitude of the selected RSS trace.
3. The leaving period experiences the *largest* decrease of RSS in the selected RSS trace.

The feature-driven scheme consists of three components: *EoL Estimator*, *Quantizer*, and *LP/BoS Estimator*. We illustrate the flow of the feature-driven scheme in Figure 5.

EoL Estimator. Assume there is a group of M segments with consecutive negative slopes in the RSS trace $\mathbb{G} = \{G_1, \dots, G_M\}$, and each segment G_i lasts for a period of $T_i = t_1^i - t_0^i$, where t_1^i and t_0^i denote the starting time and the ending time of that segment, respectively. The EoL estimator determines the time t_0^i to be EoL when a segment G_i is most likely to be a leaving period. To determine the likelihood of being a leaving period, we first define the significance of the three features described above as: 1) $\frac{T_i}{\max(\mathbb{T})}$, where \mathbb{T} denotes time length of all the segments in \mathbb{G} ; 2) $\frac{S_i}{\max(\mathbb{S})}$, where S_i is the average RSS over a time window L before t_1^i , and \mathbb{S} denotes all such average RSS; 3) $\frac{R_i}{\max(\mathbb{R})}$, where R_i is the ratio of the RSS at t_0^i and t_1^i , and \mathbb{R} denotes all such RSS ratio, respectively. We further define an utility function which is a weighted sum of the significance of the three features:

$$u_i = \alpha \cdot \frac{T_i}{\max(\mathbb{T})} + \beta \cdot \frac{S_i}{\max(\mathbb{S})} + \gamma \cdot \frac{R_i}{\max(\mathbb{R})}, \quad (1)$$

where α , β , and γ are the weights. The leaving period thus can be determined as the segment G_i that maximizes the utility function u_i . And the time t_0^i of such a segment G_i is then declared as the

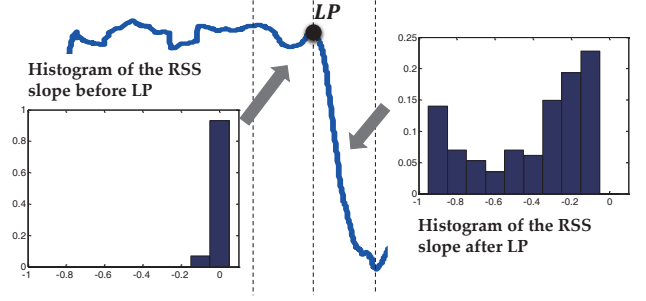


Figure 6: Histogram of the RSS slope before and after the leaving point (LP) from the RSS trace collected from a smartphone in queue.

EoL. To determine the weights in Equation (1), we use a heuristic approach by counting the occurrence of each feature in a small portion of the collected traces (e.g., 20 traces) and respectively using the ratio of each feature's occurrence to the number of traces as their weights. For example, if feature 1 has been found true for 10 times in 20 traces, the weight for feature 1 is then 0.5. Based on our experiments in different scenarios (e.g., the laboratory, the Starbucks coffee shop, and the airport), we empirically find that when $\alpha = 0.4$, $\beta = 0.8$ and $\gamma = 0.8$, the unique features can be extracted from the RSS trace accurately.

Meanwhile, we compute the slopes of each RSS sample in the selected RSS trace. The *Quantizer* performs normalization and quantization on the computed RSS slopes. The detailed implementation of the *Quantizer* is presented in Section 4.1.3.

LP/BoS Estimator. Our feature-driven scheme then takes the input from the EoL Estimator together with the quantized RSS slopes to identify the leaving point (LP) and beginning of service point (BoS) in the queue following the temporal-reversed order. The LP/BoS Estimator takes the view point that the leaving point separates the service period and the leaving period, and thus the WiFi signals before LP is relatively stable whereas they drop dramatically after the leaving point. This useful observation indicates that the distributions of WiFi signal changes (i.e., slopes of RSS) before and after the leaving point are significantly different. Figure 6 illustrates this significant difference by showing the histogram of the two RSS slope distributions before and after LP. This inspires us to identify LP by using the K-L divergence technique to examine the distribution difference of WiFi signal changes. And LP is determined as the time point (prior to EoL) separating the RSS slopes into two parts that present the most different distributions from each other.

In particular, we calculate the distributions of quantized RSS slopes before and after each time point t_j occurring before EoL, respectively denoted as $P(K_{j-1})$ and $Q(K_j)$, $j = [1, \dots, J]$, where J is number of time points occurring before EoL. The K-L divergence between these two distributions are derived as:

$$D_{KL}(P(K_{j-1})|Q(K_j)) = \sum_{q \in Q} P(K_{j-1} = q) \ln \frac{P(K_{j-1} = q)}{Q(K_j = q)}, \quad (2)$$

where Q is the set of all possible values for quantized RSS slopes. LP is then determined as the time point t_j , which maximizes the K-L divergence value:

$$t_j = \arg \max_{t_j} (D_{KL}(P(K_{j-1})|Q(K_j))). \quad (3)$$

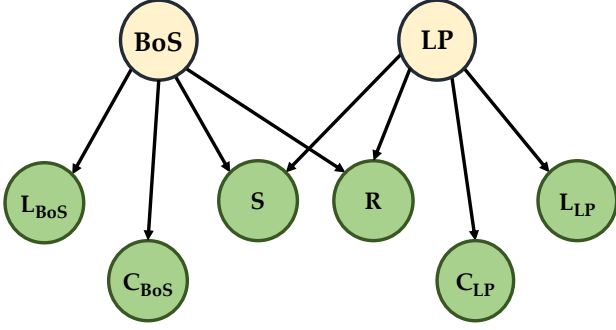


Figure 7: Bayesian Network model in our system where the random variables R , S , L_{LP} , C_{LP} , L_{BoS} , and C_{BoS} are computed from unique features embedded in the signal traces.

Similarly, our scheme can also employ the K-L divergence technique to identify BoS, since WiFi signals are relatively stable during the service period while they exhibit an obvious increasing trend in the waiting period.

3.3.2 Bayesian Network (BN) Scheme

The feature-driven scheme is straightforward and can effectively capture the unique patterns embedded in the WiFi signals when a smartphone is in a human queue. However, it does not consider the dependency among the features, which may place additional constraints to help identifying critical time points in tracking queues. In designing the Bayesian Network scheme, we use a Directed Acyclic Graph (DAG) to represent conditional independencies among the features extracted from the RSS trace. In DAG, a node represents a random variable, while an arrow that connects two nodes represents a direct probabilistic relation. Such a graph is then utilized to construct a *naive Bayesian* classifier [19] for LP and BoS determination. The Bayesian Network scheme first identifies all the RSS segments containing continuous positive RSS slopes and continuous negative RSS slopes respectively in the RSS trace and use them as inputs to run BoS and LP estimation using the constructed Bayesian classifier. The flow of the Bayesian Network approach is depicted in Figure 8.

In particular, we model BoS and LP as two parent nodes in the Bayesian Network and define six attributes (Boolean random variables) based on the features embedded in the RSS trace when a smartphone is in a human queue. The six random variables, R , S , L_{LP} , C_{LP} , L_{BoS} , and C_{BoS} , are represented as the child variables of BoS and LP in the Bayesian graphical model.

Based on the same unique features that we have shown in Section 3.3.1, we define the corresponding random variables as below:

- R : The average RSS within the service period is usually the highest. $R = 1$ when the mean RSS before a specific segment of continuous negative slopes is the highest of all, otherwise $R = 0$.
- S : The RSS within the service period is stable. $S = 1$ when the variation of RSS within a window W before a segment of continuous negative slopes is smaller than a threshold τ , otherwise $S = 0$.
- L_{LP} : The leaving period happens after the service period, which has the strongest RSS. $L_{LP} = 1$ when the starting time of a specific segment of continuous negative slopes is later than the time with the highest RSS value, otherwise $L_{LP} = 0$.

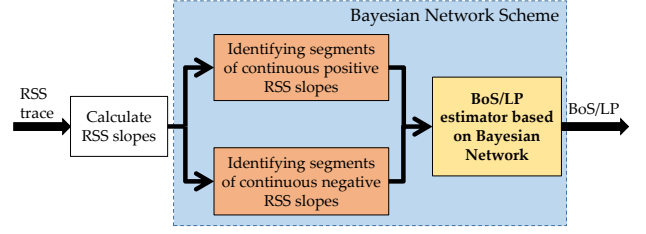


Figure 8: Flow of the Bayesian Network scheme to find BoS and LP.

- C_{LP} : The change of RSS after the leaving period usually exhibits the most significant decreasing trend. $C_{LP} = 1$ when the average slope of a specific segment of continuous negative slopes is the smallest of all, otherwise $C_{LP} = 0$.
- L_{BoS} : The waiting period happens before the service period, which has the strongest RSS. $L_{BoS} = 1$ when the end time of a specific segment of continuous positive slopes is earlier than the time with the highest RSS, otherwise $L_{BoS} = 0$.
- C_{BoS} : The change of RSS before the service period usually exhibits the most significant increasing trend. $C_{BoS} = 1$ when the average slope of a specific segment of continuous positive slopes is the largest of all, otherwise $C_{BoS} = 0$.

Assume the scheme identifies a set \mathbb{U} of N segments with continuous positive slopes, and a set \mathbb{G} of M segments with continuous negative slopes in the RSS trace. The naive Bayesian classifier considers the ending time of the segment $U_i \in \mathbb{U}$ with continuous positive slopes as BoS when the segment U_i maximizes the posterior probability:

$$\hat{U} = \arg \max_{U_i} (P(BoS) \prod_{I \in \mathbb{I}} P(I = 1 | BoS)), \quad (4)$$

where $\mathbb{I} = \{R, S, L_{BoS}, C_{BoS}\}$, and $P(BoS) = \frac{1}{N}$. Similarly, the starting time of the segment $G_i \in \mathbb{G}$ with continuous negative slopes is considered as LP when the segment G_i maximizes the posterior probability:

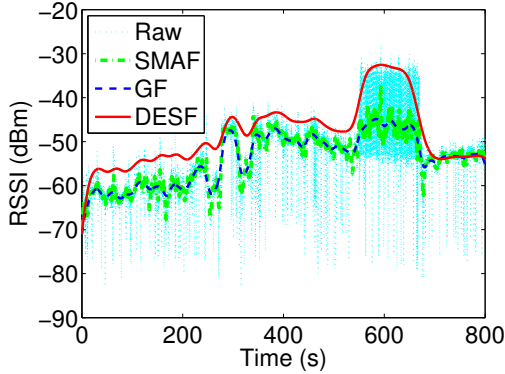
$$\hat{G} = \arg \max_{G_i} (P(LP) \prod_{F \in \mathbb{F}} P(F = 1 | LP)), \quad (5)$$

where $\mathbb{F} = \{R, S, L_{LP}, C_{LP}\}$, and $P(LP) = \frac{1}{M}$.

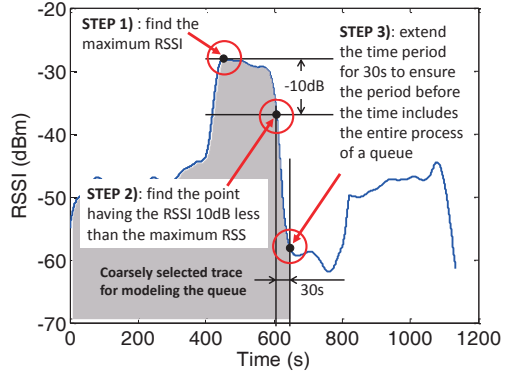
We adopt the traditional two-fold cross-validation approach [12] when applying the Bayesian Network scheme. We randomly separate the collected RSS traces from multiple smartphones in a queue into two folds and use one fold as the training data to learn the probability of each random variable conditioning on its parent variable. The other fold is used as the testing data to obtain the estimation error. Then we switch these two folds and calculate the estimation error again. The final estimation error of the system is the average of the estimation errors observed from previous two tests.

4. IMPLEMENTATION AND LAB EXPERIMENTS

In this section, we present *Data Calibration* and *Signal Integration* sub-tasks in our system. We then describe two approaches to track the important time periods of human queues: *single/multiple antenna* and *single/multiple user*. Furthermore, we provide an evaluation on different queue patterns and energy consumption analysis of our approach.



(a) Different filters



(b) Coarse queue identification

Figure 9: Data calibration: (a) comparison of results from the simple moving average filter (SMAF), the dynamic exponential smoothing filter (DESF) and the gaussian filter (GF); (b) illustration of Coarse Queue Identification: identifying the RSS segment containing the entire process of a human queue.

4.1 Data Calibration to Maintain Signal Reliability

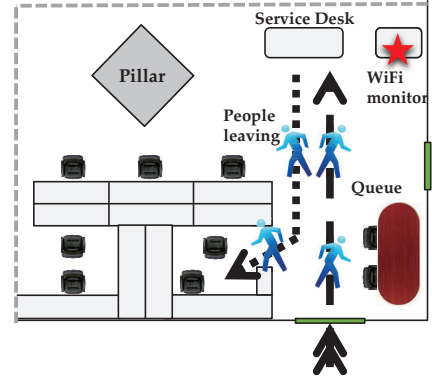
Although the RSS value is governed by the distance between the smartphone and the monitor, the RSS readings are quite noisy due to the multi-path, shadowing, and fading components of a signal as people move. They are also affected by the way people hold the phones. Further, the collected RSS traces may contain irrelevant time periods, for example, the period after people leave the queue. The *Data calibration* sub-task aims to clean the noisy data, identify the relevant signal segment containing the queue process, and preserve the unique patterns in the signal trace.

4.1.1 Data Interpolation

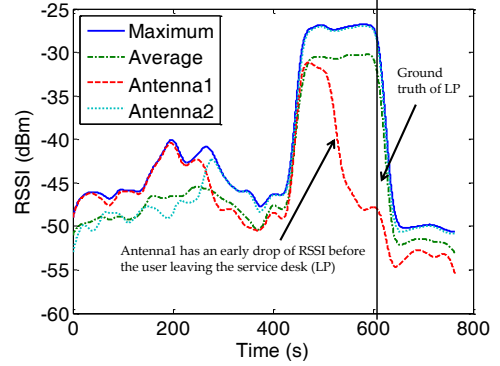
This step is used to interpolate the RSS samples in evenly spaced time series as we observe that even with an app broadcasting beacon packets periodically, the interval between two adjacent RSS samples has a small variation due to packet delay. Given a sampled RSS trace starting at t_0 , we interpolate the RSS trace at the time points $[t_0, t_0 + \Delta, t_0 + 2\Delta, \dots]$, where Δ is the interpolation step. In our experiments, we find an interpolation step of $0.1s$ is small enough to produce sufficient samples for queue parameter determination.

4.1.2 Low-pass Filtering

Low-pass filtering aims to remove high frequency noise (due to environment dynamics) and yet capture the statistical features pre-



(a) Experimental setup in the lab



(b) Compare single/multiple antennas

Figure 10: Queue measurement in the laboratory: (a) illustration of Experimental setup; (b) comparison of RSSI traces from two individual antennas and the integrated RSSI trace using both antennas.

sented in the trace. We experiment with three low-pass filters: simple moving average filter (SMAF), dynamic exponential smoothing filter (DESF) and Gaussian filter (GF). Figure 9 (a) shows the raw RSS trace and the resulted trace after applying these three filters. We observe that both SMAF and GF preserve the rough trend of the trace but fail to capture the exact time points when people arriving at the service desk and moving away after got service due to they has slow roll-off and poor stopband attenuation. We find that DESF not only returns the nicely fitted curve but also preserves those critical points in the RSS trace. This is because DESF is an exponential smoother that changes its smoothing factor dynamically according to previous samples. We thus choose the DESF in our system to remove high frequency noise while preserving the features in the RSS trace. Specifically, we implement DESF based on following equation:

$$s_i = \begin{cases} \alpha \cdot s_{i-1} + (1 - \alpha) \cdot x_i, & \text{if } x_i < s_{i-1} \\ (1 - \alpha) \cdot s_{i-1} + \alpha \cdot x_i, & \text{if } x_i \geq s_{i-1}, \end{cases} \quad (6)$$

where s_i is the output sample and x_i is the input sample, s_{i-1} is the previous smoothed sample, α is the smoothing factor that we can choose from $(0.5, 1)$ to favor the samples with larger values.

4.1.3 Coarse Queue Identification

This step is to identify the signal segment that belongs to the human process queue as the collected trace may include the periods that people have left the queue already but still within the monitoring range of the WiFi monitor. The irrelevant time periods con-

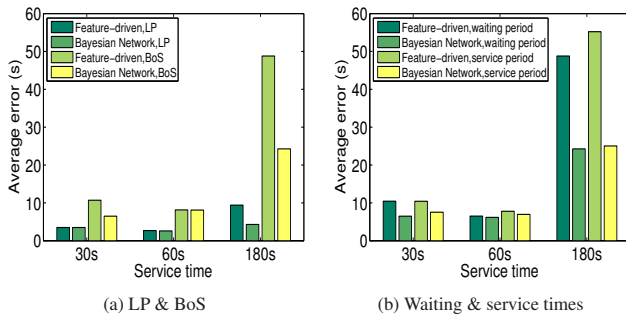


Figure 11: Single-user and single-antenna approach: performance comparison of the feature-driven and Bayesian Network schemes under different service times in laboratory.

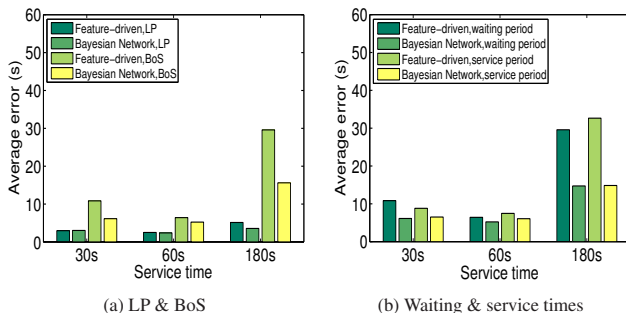


Figure 12: Single-user and two-antenna approach: performance comparison of the feature-driven and Bayesian Network schemes under different service times in Laboratory.

tain useless information and may even affect the accuracy of queue parameter determination due to unpredictable behaviors of people after leaving the queue.

We perform three steps to remove irrelevant time periods: 1) find the time point having the maximum RSS of the entire trace; 2) examine the RSS after that time point to locate the time when the difference between the RSS value and the maximum RSS is larger than a threshold, for example $-10dB$, which is large enough to fall into the leaving period, and 3) remove the time periods after a time interval (e.g., the leaving period is usually within 30s) on top of the time point we identified in step 2 to ensure that the leaving time period is not excluded. We illustrate these three steps in Figure 9 (b).

Quantizer. After coarse queue identification, we apply a first order linear regression to generate the slope for the RSS segment centered at each RSS sample. The resulting RSS slopes are then normalized and linearly quantized before input into the queue parameter determination algorithms (i.e., feature-driven or Bayesian Network schemes). For normalization and quantization, we use 11 negative levels between $[-1, -0.9, \dots, -0.1, 0]$ when estimating LP and 11 positive levels between $[0, 0.1, \dots, 0.9, 1]$ when estimating BoS.

4.2 Inference Using Single and Multiple Antennas

We next show our system implementation with a single antenna as well as two antennas at the WiFi monitor, and evaluate their impacts.

We first conduct experiments with human queues in a laboratory setup as shown in Figure 10 (a). We present the extensive evaluation of our system in two public facilities in Section 5. We place

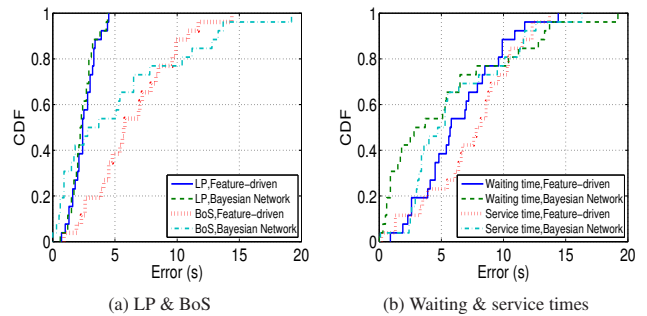


Figure 13: Single-user and two-antenna approach: error CDF using the feature-driven and Bayesian Network schemes under the short (60s) service time in Laboratory.

a WiFi monitor equipped with two antennas at the service desk to capture WiFi signals. An Android app is developed to send beacon packets at the rate of $10pkt/sec$ to simulate the normal WiFi traffic, such as the traffic generated by the Starbucks mobile app. We experiment with three different service times, 30s, 60s, and 180s, representing *short*, *normal* and *long* service times, respectively. In total we collect 90 traces in the laboratory environment and we use two-fold cross-validation for evaluation. We calculate the estimation error of the relevant time points and important time periods with regard to the manually logged ground-truth. In our experiments, we have both line-of-sight and non-line-of-sight scenarios, since the signals from smartphones in most cases are blocked by human bodies in the queue.

4.2.1 Single Antenna Approach

We first study the performance of the feature-driven and Bayesian Network schemes based on the signals received from a single antenna at the WiFi monitor. Because the leaving period is a brief period that facilitates the derivation of the service period, in the rest of this paper, we will focus on presenting the results of estimating waiting and service periods. Figure 11(a) compares the average error of estimating the LP and BoS under different lengths of the service time. We observe that both the feature-driven and BN schemes perform well under short (30s) and normal (60s) service times with the average error less than 10s. We find that under the long service time (180s), the BN scheme outperforms the feature-driven scheme. Moreover, Figure 11(b) depicts the average error of tracking waiting and service times derived from the estimated BoS and LP. We observe the average error of both waiting and service times is less than 10s under short and normal service times. Under the long service times, the BN scheme also outperforms the feature-driven scheme by about 50%, although both schemes experience performance degradation. This is because the BN scheme considers the conditional relationship among multiple features and thus places a tighter bound on the critical queue points estimation even when large RSS fluctuation is experienced during a long service time period.

4.2.2 Multiple Antenna Approach

When multiple antennas are available, integration of signals from them may help to mitigate the multi-path effects and make the wireless signal reception robust to environment dynamics. It is common for a WiFi monitor (e.g., WiFi access point) to have two antennas. We perform a study of integrating the signals from two antennas of the WiFi monitor in our laboratory using two strategies: 1) *Maximum*: taking the maximum RSS from two antennas, 2) *Average*: averaging RSS from two antennas. Figure 10 (b) compares the RSS

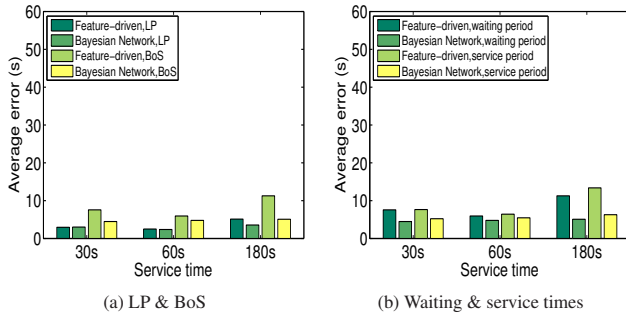


Figure 14: Two-user and two-antenna approach: performance comparison of the feature-driven and Bayesian Network schemes under different service times in Laboratory.

trace from each antenna and the resulted RSS trace after combining from two antennas applying the two strategies. We find that although the RSS from the Antenna 2 drops quickly before reaching LP (affected by multi-path effects and dynamically changed environments), the integrated RSS traces, both *maximum* and *average*, accurately reflect each critical time point in the queue.

Figure 12 shows the average error of BoS and LP estimation and the derived waiting and service times in the laboratory using the *maximum* signal integration strategy (Note that the *average* strategy results in similar performance). Comparing to single antenna results shown in Figure 11, the average error in both Figure 12 (a) and (b) has been reduced over 30% under the long service time, indicating the integration of multiple antennas is highly effective in reducing large errors caused by signal fluctuations. Furthermore, Figure 13 depicts the cumulative distribution function (CDF) of the estimation error of the related time points and time periods in queue under the normal service time. The results confirm that the estimation error of LP is less than 4s for 80 percentile and the maximum error for time periods estimation is less than 20s.

4.3 Inference Using Multiple Users

With the increasing usage of smartphones, it is common to have back-to-back smartphones presenting in the queue. The crucial time points, e.g. BoS and LP, of such back-to-back smartphones have strong correlations. The BoS of the subsequent smartphone happens very soon after the LP of the prior neighbor smartphone. This kind of correlation among neighboring phones can contribute to reducing estimation uncertainties when tracking human queues.

In particular, we observe that the LP estimation error is usually 50% smaller than that of BoS, suggesting that we can leverage the LP estimation of the prior phone as a reference to bound the BoS estimation of the subsequent phone in the queue. Toward this end, we develop a simple yet effective process after estimating BoS of the subsequent smartphone. We compare the estimated BoS of the subsequent smartphone to the estimated LP of its prior neighbor phone: if the difference is larger than a threshold θ , we use the LP of the prior smartphone plus θ as the estimated BoS of the subsequent phone, otherwise we consider the estimation of BoS is accurate and keep it. Theoretically the LP of prior phone should be the same as the BoS of the subsequent phone. However due to the delay of people's movement to the service desk, there could be a time difference between these two. In our approach, we utilize this time delay to determine the θ , which is about 5s empirically.

Figure 14 shows the two-antenna results when using the correlation between two smartphone users. Comparing to the BoS estimation errors when using a single user's RSS trace (in Figure 12), we observe a 25% improvement in BoS estimation under the short

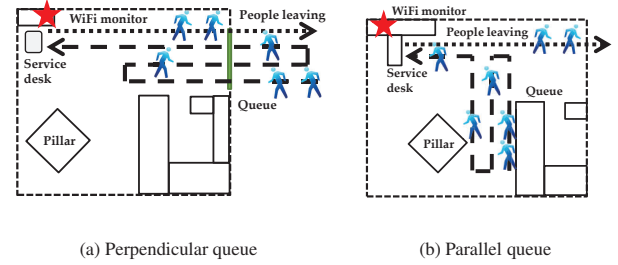


Figure 15: Two cases of the snaking queue in laboratory.

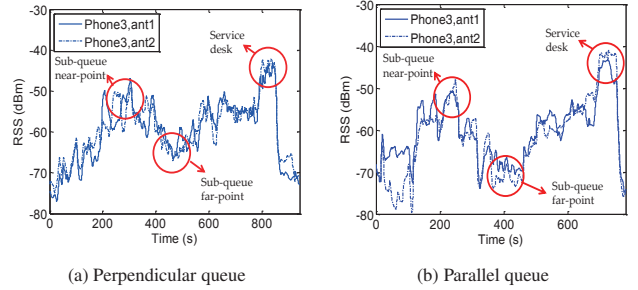


Figure 16: Example: RSS traces of one smartphone user in two cases of the snaking queue show clearly up-down trends corresponding to the positions of the user in queues.

and normal service times and about 60% improvement under the long service time. In addition, we observe the similar improvement in waiting and service times estimation. This encouraging observation confirms that leveraging the correlation among multiple users in the queue can help reduce large errors when tracking the important time periods in human queues with a single-point signal monitor.

4.4 Evaluation on Snaking Queues

In many scenarios, queues are formed not only in a straight line, but also in folding or zig-zag patterns with multiple subqueues. It is thus important to evaluate the flexibility of our schemes in handling queues with different patterns such as folding or zig-zag queues. We consider a general snaking queue consisting of subqueues with multiple straightlines. We study two representative cases of snaking queues in the laboratory environment: *perpendicular queue* and *parallel queue*. The moving direction of the subqueues in the *perpendicular queue* is perpendicular to the service desk, while they are in parallel to the service desk in the *parallel queue*. The experimental setup of these two queue patterns are illustrated in Figure 15(a) and (b) respectively. The same single monitor is utilized to collect WiFi signals from smartphones. We leave the evaluation in real environments to the next section.

Figure 16 depicts the signal traces collected from these two cases of snaking queues during our experiments. We observe clear and slow up-down trends in the signal trace when the user moves from one subqueue to another in both types of queues. After the user receives the service and leaves the queue finally, a sharp signal drop is observed just as in the single-line queue. Additionally, Figure 17 shows the average estimation error in waiting and service times of snaking queues when the service time is around 60s and the waiting time is around 500s. We utilize the maximum signal integration strategy to combine readings from two antennas before the queue parameter determination. We observe that our feature-driven and Bayesian Network schemes can achieve low estimation errors in

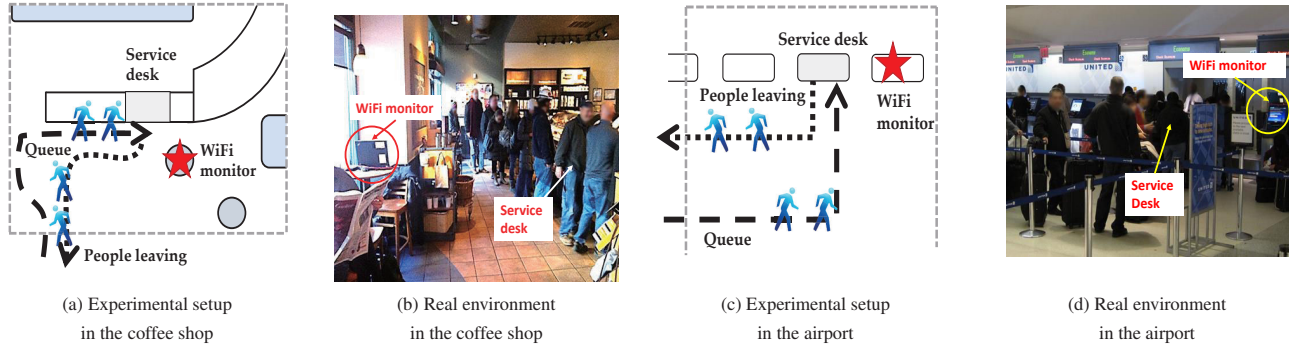


Figure 18: Illustration of experimental setups and real environments in a coffee shop (a and b) and an airport (c and d).

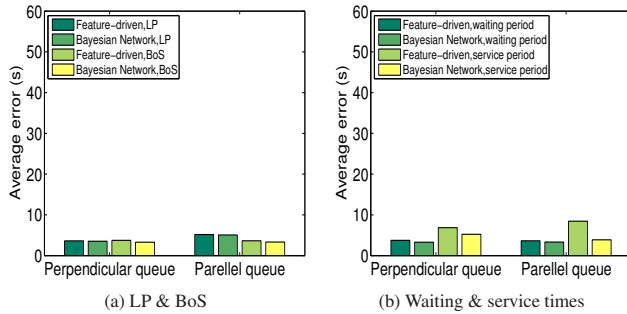


Figure 17: Evaluation of snaking queues in laboratory using single-user and two-antenna approach: feature-driven and Bayesian Network schemes.

both types of snaking queues. Specifically, the average LP and BoS estimation errors are less than 6s, and the average errors of waiting and service times estimation are about 4s and 8s, respectively. These results are comparable to the ones obtained in single-line queues. This is because our schemes utilize a time-reversed manner to examine the signal traces and the unique features embedded in the signal traces (e.g., the largest signal drop is observed when the user leaves the service desk; the highest and stable signal values are encountered when the user is receiving the service) are still preserved even though the RSS goes up and down when the users go through the snaking queues.

4.5 Energy Consumption Analysis

Our approach require the WiFi interface remains alive during the queue process, we estimate the energy overhead of our approach by utilizing the tool and methodology in [25]. The main power consumption component on the smartphone is an app (e.g., customer loyalty app or a customized app) periodically broadcasting beacons through WiFi. We find that sending these broadcasting beacon packets for 5mins (which is the average length of RSS traces that include the entire queue process) with 10pkt/s rate takes about 54.3 Joule, which is about 180mW. Furthermore, with a reduced packet sending rate at 5pkt/s, our approach can achieve similar estimation accuracy while the power consumption keeps as low as 80mW. This is much smaller than the average power consumption of a phone (e.g., HTC Evo 4G lasts 12.7 hours with average power of 450mW). Therefore we conclude that our approach does not pose a burden for the battery life of smartphones.

5. PROTOTYPE EVALUATION IN REAL ENVIRONMENTS

In this section, we evaluate our system of tracking human queues in two real environments with three types of smartphones.¹

5.1 Experimental Scenarios

To evaluate our system, we conduct experiments in a coffee shop and an airport (a check-in line for domestic flights with both automated machine and a human representative) for over 20 hours and 6 hours, respectively. During the experiments, we place a WiFi monitor close to the service desk in the coffee shop and the check-in desk in the airport. In each round of the experiment, the participants turn on our app to generate WiFi traffic when they get into the queue. Our system tracks the human queues waiting for service in both the coffee shop and the airport as illustrated in Figure 18. The WiFi monitor is made of an Alix system including two 100M Ethernet interfaces and two Wistron NeWeb CM9 802.11a/b/g-based WiFi radio modules connecting to two separated antennas. The Alix system runs on a modified Linux OS that allows running customized software to collect and process data. We developed a java program to let the Alix system log RSS traces based on phones' MAC address in the queue. We note that only RSS and the corresponding time stamps are preserved for offline study. The program can fetch the RSS trace per MAC address through the Ethernet connection to further track important time periods of the queue. To validate our system, we do not need to involve many participants inside the same queue. For example, we only place 5 participants in the service queue with the queue length ranging from 7 to 12 people and track the signal changes of their smartphones. These smartphones include HTC 3D, HTC EVO 4G, and Nexus One. In total we collected 72 traces in the coffee shop and 54 traces in the airport. Our scenarios have included both line-of-sight and non-line-of-sight scenarios, and basically the non-line-of-sight scenarios are the majority of the cases. The queue in the coffee shop experiences a normal service time of about 60s, while the queue in the airport presents a longer average service time around 180s. We calculate the estimation error with regard to the manually logged ground-truth of the waiting, service, and leaving time.

5.2 Evaluation in a Coffee Shop

We first evaluate the effectiveness of our system in a coffee shop. Figure 19 presents the cumulative distribution function (CDF) of estimation errors in the coffee shop when using the signal integration of two antennas on single user traces. In general, Bayesian Network scheme achieves better accuracy than the feature-driven

¹This project has obtained IRB approval.

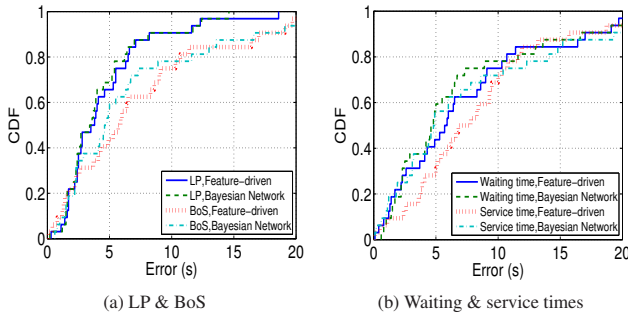


Figure 19: Coffee shop using single-user and two-antenna approach: error CDF using the feature-driven and Bayesian Network schemes with normal service time around 60s.

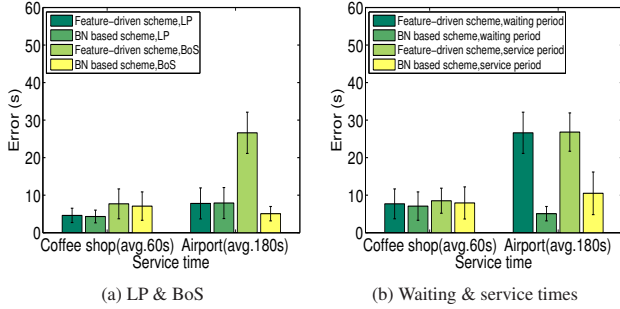


Figure 20: Coffee shop and airport with single-user and two-antenna approach: performance comparison of the feature-driven and Bayesian Network schemes.

scheme. Particularly, Figure 19(a) shows that the median error of LP and BoS estimation is about 4s and 5s respectively for both schemes. And the median errors of the derived waiting and service time periods are less than 7.5s as shown in Figure 19(b). Given the ground-truth of the average service time, the estimation error of the service period is only about 11%. This indicates that our system can track human queues with high accuracy by using only a single-point signal monitor and a sample of phones in the queue.

5.3 Coffee Shop and Airport Performance Comparison

Next, we compare the performance of our system in a coffee shop and an airport with average service time around 60s and 180s respectively.

Feature-driven scheme. Figure 20(a) shows that the average LP estimation error using feature-driven scheme is low in both environments, around 5s and 7s for the coffee shop and airport respectively. However, we observe that the average error of BoS estimation in the airport increases, which results in higher average errors on estimating the waiting and service times as shown in Figure 20(b). This error is about 15% of the average service time obtained from the ground-truth, which is a small portion of the total service time and is still under the acceptable range. In addition, this observation is also compatible to the experimental results obtained in the laboratory environment with the long service time. We also observe that the standard deviation of the estimation errors is lower in the coffee shop (around 4s) and about 8s in the airport.

Bayesian Network scheme. Comparing to the feature-driven scheme, the Bayesian Network scheme provides consistent low estimation errors in both environments: The LP and BoS estimation errors are less than 8s for both coffee shop and airport. Whereas

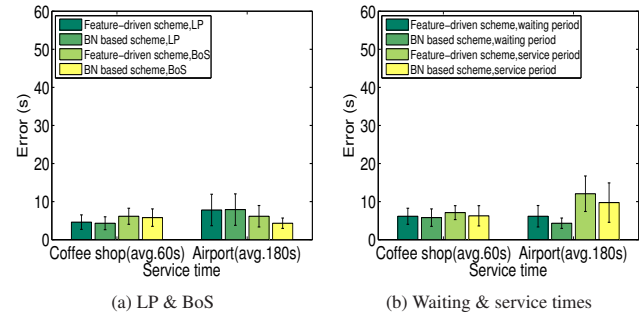


Figure 21: Coffee shop and airport, two-user and two-antenna approach: performance comparison of the feature-driven and Bayesian Network schemes with different service times.

the waiting and service times estimation achieves low average errors less than 8s in coffee shop and about 5s and 10s in the airport, respectively. The standard deviation of estimation errors in both environments is around 3s to 7s. Because passengers in the airport have to pause at a stop line that is about 7 feet from the service desk before they are allowed to proceed to the service desk, this behavior contributes to a more obvious increasing RSS trend when the passenger goes up for service and helps the Bayesian Network scheme maintains its performance to identify the service periods accurately under long service times. We conclude that the Bayesian Network scheme is more robust to real environments but with higher computational cost than the feature-driven scheme.

5.4 Leveraging Multiple Users

Furthermore, we leverage the presence of multiple users in a queue to improve the BoS estimation in both real environments. We follow the same procedure as described in Section 4.3. To make a fair comparison, we apply the same threshold as that in laboratory experiments to both real environments, i.e., $\theta = 5s$. Figure 21 (a) shows that by utilizing the LP estimated from the RSS trace of the prior smartphone, the BoS estimation error of the subsequent smartphone is significantly reduced when using the feature-driven scheme. Comparing to Figure 20, the average BoS estimation error in the airport has about 50% improvement, which is compatible to the results when the Bayesian Network scheme is applied. Additionally, Figure 21 (b) shows that the estimation error of waiting and service times in the airport also have about 50% improvement. Both feature-driven and Bayesian Network schemes reduce the error to 5s and 10s for waiting and service periods, respectively. The standard deviation of the estimation error is also reduced to about 2s in the coffee shop and 6s in the airport. This result suggests that employing two consecutive smartphone users (when available) in the queue is helpful to make our system more robust under real-world scenarios.

5.5 Evaluation with Snaking Queues

Finally, to evaluate the performance of our system under snaking queues in real environments, we find a second coffee shop with the presence of a snaking queue as illustrated in Figure 22. We place the WiFi monitor on the service desk, which is about 5m away from the nearest point in the sub-queue, and the average service time and waiting time in the coffee shop is about 60s and 290s, respectively. There are about 40 people in the store. Figure 23(a) shows that the average estimation error of the waiting and service times are both less than 10s for both the feature-driven and Bayesian Network schemes. In addition, Figure 23(b) presents the CDF of the errors of waiting and service time estimation. It is encouraging that the

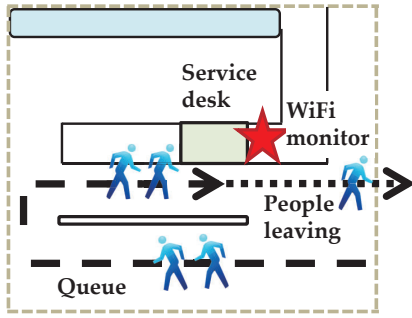


Figure 22: Illustration of experimental setup in a second coffee shop with the presence of a snaking queue.

results are comparable to those with single-line queues, indicating that our queue parameter estimation system is flexible and scalable to queues with different patterns.

6. DISCUSSION

The results show that our WiFi-based solution can measure different kinds of queues, especially longer queues, with only a single monitoring station. Earlier camera solutions are prone to occlusion and may require multiple cameras (networked together) to provide a complete measurement. Multi-camera setups are more costly in hardware and installation and may encounter privacy concerns because customers' facial identities are captured by the cameras. There are still, however, some remaining challenges and opportunities for future work that we discuss below.

Opportunistic WiFi Traffic. Additional work may be needed to generalize the system to other WiFi traffic patterns, especially when regular beacons are not available. Our current experiments relied on an app that generates periodic WiFi packet transmissions. While it should be possible to add such functionality to apps that are frequently used in certain queues (e.g., frequent customer apps or airline checkin apps), it would also be desirable for this system to opportunistically use existing WiFi traffic from persons in a queue such as traffic from watching videos, streaming music, checking email, or reading news. Some of these traffic sources are likely to be more sporadic and it is possible that the traffic from a phone stops while the person is still waiting in the queue which poses additional challenges for estimating queue parameters accurately. There are a couple of possible solutions to address these problems: (1) the system can filter out traces that do not possess the sharp drop signal pattern associated with leaving the queue, because this trace is likely incomplete; (2) the system could aggregate multiple signal traces from different users in the queue to reduce the uncertainty of the queue parameter estimation. Although a single phone user in the queue may not be sufficient, the system may still be able to estimate the queue parameters accurately with the help from multiple smartphone users in the queue.

Existing WiFi Access Points. An alternative to deploying a dedicated monitoring device is to utilize existing access points (APs) to capture the WiFi traffic and perform queue measurements. While attractive because the use of multiple APs could improve accuracy, this requires appropriate interfaces to obtain station RSS traces and the placement of existing APs may not be ideal to detect the features we have used in this system. In particular, existing APs are likely placed to maximize coverage in higher locations and may not be close to the service desk of the queue. This means that the signal drop after service can be less pronounced. Still, when appropriate

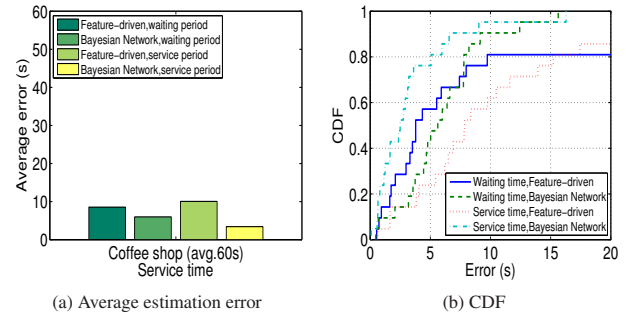


Figure 23: Performance under a snaking queue in a second coffee shop using single-user and two-antenna approach: statistics of estimation errors of waiting and service times.

RSS interfaces are available, this queue monitoring system could potentially make use of existing WiFi infrastructures.

Queue Patterns. Although we show that our queue parameter estimation system can cover a broad range of scenarios, there still exists queues that can present challenges. One are disorganized queues such as those at bus stops or train-station platforms, where the service time is short (e.g., people getting on the bus) and there are no clear first-in and first-out (FIFO) service patterns. Other challenging queues are those in relatively small spaces without pronounced leaving patterns after the service. In such cases, additional or different features need to be identified for integration into our feature-driven and Bayesian Network schemes. For example, when dealing with disorganized queues, new unique features could include a constant observation of multiple mobile devices with stable or small-varying signals for a long period producing a temporal cluster (e.g., people waiting at the bus stop) followed by near-simultaneous departures (e.g., people getting on the bus and departing). These new features will help to estimate critical queue parameters.

7. RELATED WORK

The prevalence of wireless technology and smartphone sensors has enabled great convenience on measuring people's group behavior in daily life. The habitual nature of human mobility in wireless networks have been well investigated [14, 15, 20]. For example, Kim and Kotz analyze users' mobility characteristics including pause time, speed, and direction of movements based on the WiFi traces [14]. And many works [18, 22] recognize human activities (for example walking, sitting, talking, running) by exploiting the readily available sensing capabilities on smartphones such as audio and accelerometer sensors. These works indicate it is feasible to study human group behaviors using existing wireless technology and smartphones.

Turning to examining people staying time in retail stores and business areas, Manweiler et al. [17] designs a coarse-grained dwell-time prediction framework in retail store environments by leveraging the accelerometer and gyroscope sensor readings and WiFi signals from smartphones. Ghosh et al. [8] determines arrival counts and temporal variations, which can be utilized to model the connection time of a user, by modeling the characteristics of traffic observed at a large number of public WiFi hotspots. An anonymous Bluetooth tracking system is deployed at the Indianapolis International Airport to measure the time of passengers staying at security checkpoints [3]. However, none of above works provide detail information to track human queues.

As a particular form of group behavior, queuing has also been studied for many years. The broad utility of queuing in real en-

vironments could benefit both service providers as well as customers to adjust their behavior and process accordingly. However, the assumptions of classical queueing theory may not be applicable to model real-world situations exactly [9]. Existing solutions to the queue monitoring problem often rely on additional infrastructures with multiple sensors with high costs and maintenance overheads. For example, the Swedish pavilion at the 2010 World Expo in Shanghai uses multiple video cameras to count people in queues [1] and sensors including light barriers, switching mats, and Bluetooth are exploited to measure the travel and service times in airports [2, 3]. Infrared sensor technique also provides a solution to model vehicle queues on highways [24]. Furthermore, there are several works utilizing virtual queue system to monitor and provide queue information in theme parks [5] and hospitals [6]. But the virtual queue system cannot track human queues in real time. The novelty of our work is that we explore to use a single-point signal monitor to track human queues leveraging existing WiFi traffic from smartphones in queue. To the best of our knowledge, this work is the first to provide a low infrastructure solution that can accurately estimate queue parameters.

8. CONCLUSION

In this paper, we present a low-infrastructure approach for tracking human queues by extracting unique WiFi signal patterns from smartphones in queues. The approach only requires a small fraction of people in the queue using WiFi on their smartphones and in contrast to earlier solutions needs only a single signal monitor. The insight that the signals from smartphones in a queue follow a predictable pattern as they move through the queue not only allows us to monitor the total time spent inside the queue but also distinguish different phases such as, waiting, service and leaving periods, and estimate the parameters in a queueing model. We propose two core schemes to recognize critical time points that separate these phases: one directly uses special signal features, and the other is based on a Bayesian Network model that considers not only the special features, but also the relation between them.

We demonstrate the generality of our approach applied to a variety of queue patterns through extensive experiments including laboratory, coffee shop, and airport. We observe that our system yields only about 5s estimation error when service time is short and normal (about 30s and 60s). Even with service time as long as 180s, our approaches can achieve the estimation error less than 10s, which is accurate comparing to the length of service time. Although we only experimented with a first-in first-out queueing process, most other processes still fit into our simplified model, because we leave the service order undefined.

9. ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation Grants CNS1217387, CCF1018270, CNS1040735, and CNS 0845896.

10. REFERENCES

- [1] MultiQ at Shanghai World Expo 2010. http://www.multiq.com/wpcontent/uploads/2013/05/case_study_shanghai_world_expo_2010.pdf.
- [2] D. Bauer, M. Ray, and S. Seer. Simple sensors used for measuring service times and counting pedestrians. *Transportation Research Record: Journal of the Transportation Research Board*, 2214(1):77–84, 2011.
- [3] D. M. Bullock and et.al. Automated measurement of wait times at airport security. *Transportation Research Record*, 2010.
- [4] G. Chandrasekaran and et.al. Vehicular speed estimation using received signal strength from mobile phones. In *ACM Ubicomp*, 2010.
- [5] R. F. Cope III, R. F. Cope, and H. E. Davis. Disney’s virtual queues: A strategic opportunity to co-brand services? *Journal of Business & Economics Research*, 2011.
- [6] D. Dickson, R. C. Ford, and B. Laval. Managing real and virtual waits in hospitality and service organizations. *Cornell Hotel and Restaurant Administration Quarterly*, 2005.
- [7] S. Dimatteo, P. Hui, B. Han, and V. Li. Cellular traffic offloading through wifi networks. In *IEEE MASS*, 2011.
- [8] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, and N. Shankaranarayanan. Modeling and characterization of large-scale wi-fi traffic in public hot-spots. In *IEEE INFOCOM*, 2011.
- [9] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley-Interscience, 2008.
- [10] D. Hanchard. FCC chairman forecasts wireless spectrum crunch. <http://www.zdnet.com>, 2010.
- [11] R. Handford. Starbucks hits more than 10m active app users. <http://www.mobileworldlive.com/starbucks-hits-more-than-10m-active-app-users>, 2013.
- [12] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.
- [13] S. Isaacman and et.al. Human mobility modeling at metropolitan scales. In *ACM MobiSys*, 2012.
- [14] M. Kim and D. Kotz. Extracting a mobility model from real user traces. In *IEEE INFOCOM*, 2006.
- [15] B. Liang and Z. J. Haas. Predictive distance-based mobility management for multidimensional PCS networks. *IEEE/ACM Transaction on Networking*, 2003.
- [16] I. I. S. Ltd.". Queue management. <http://www.irisys.co.uk/queue-management/>, 2013.
- [17] J. Manweiler, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Predicting length of stay at wifi hotspots. In *IEEE INFOCOM*, 2013.
- [18] E. Miluzzo and et.al. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *ACM SenSys*, 2008.
- [19] A. Mittal and A. Kassim. *Bayesian network technologies: applications and graphical models*. IGI publishing, 2007.
- [20] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *ACM MobiCom*, 2008.
- [21] I. of Medicine (US). *Hospital-Based Emergency Care: At the Breaking Point*. National Academy Press, 2007.
- [22] M. Rabbi, S. Ali, T. Choudhury, and E. Berke. Passive and in-situ assessment of mental and physical well-being using mobile sensors. In *ACM UbiComp*, 2011.
- [23] I. Siemens Industry. Measure travel time and speed between multiple controller locations. <http://www.mobility.siemens.com>, 2011.
- [24] U. Stilla, E. Michaelsen, U. Soergel, S. Hinz, and H. Ender. Airborne monitoring of vehicle activity in urban areas. *ISPRS*, 2004.
- [25] L. Zhang and et.al. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *ACM CODES/ISSS*, 2010.