Immediate Cooperative Line Wait Time Estimation System Using BLE on Smartphone

Yujin Miyagawa and Masakatsu Ogawa

Graduate School of Science and Technology, Sophia University
7-1 Kioicho, Chiyoda-ku, Tokyo 102-8554, Japan
E-mail: yujin-6932@eagle.sophia.ac.jp, m-ogawa@sophia.ac.jp

Abstract

In this paper, we propose an immediate cooperative line wait time estimation system using Bluetooth low energy (BLE, marketed as Bluetooth Smart) on a smartphone; this system is a modified version of our previous proposed method. To estimate the wait time, we utilize time stamps of when users approach and move past two preinstalled receivers. Our system comprises three main components: the receivers, a wait time estimator, and a database. The receivers record two types of data: the recorded time and the RSSI values. The wait time estimator uses the wait time estimation algorithm, which includes three main subroutines: the maximum RSSI decision, in-decision, and out-decision on the receivers for each user’s smartphone. By calibrating and analyzing the recorded log data, the wait time estimator estimates the estimated wait time. This estimated wait time is stored in the database, then provided through the website to the queueing users. The experimental results showed that the difference between the estimated wait time and the expected wait time was within 10 s for all measurements.

1. Introduction

People often have to wait in a queue at facilities, such as food courts, banks, and amusement parks. In [1], Houston et al. proposed that there is a close relationship between the wait time and the user’s evaluation of service quality. To solve this problem, facilities have focused on estimating and displaying the wait time during busy periods in real time, although it is undesirable to apply costly special devices, such as monitoring cameras or infrared sensors.

Smartphones equipped with a Bluetooth low energy (BLE) specified in Bluetooth 4 are now commonly used. iBeacon specified by Apple Inc. uses BLE and specifies its packet format in the BLE advertisement packet.

In this paper, we propose an immediate cooperative line wait time estimation system by utilizing the analysis of BLE broadcasting transmitted from a smartphone application. This system is a modified version of our previous proposed method, with which we could not estimate the wait time immediately or provide it to queueing users [2].

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 describes our immediate estimation system. Section 4 describes our experiments and results. Conclusions are presented in Sect. 5.

2. Related Works

Real-time estimation of the wait time in human queues is useful for optimizing processes across various service industries. Managers, service providers, and even customers can immediately change their behavior and processes as needed.

For example, there are several examples of utilizing a virtual queue system to monitor and provide human queue information including the wait time in amusement parks [3] and similar locations. However, these systems cannot estimate the wait time in human queues immediately.

In [4], Bulut et al. described LineKing, a crowdsourced line wait time estimation system. In this system, user’s smartphones require connection and data communication with a cloud server. Our system utilizes BLE broadcasting by smartphones; thus, there is no requirement for these processes between smartphones and a server. In [5], Wang et al. described a tracking human queues system using single-point Wi-Fi signal monitoring. They define a three periods in a human queue: waiting, service and leaving periods. This system aimed to estimate the end points of these periods except for the start point of the waiting period. Our system estimates the start and end points of waiting period; therefore we can obtain higher accuracy for the estimated wait time.

3. Immediate Estimation System

3.1 Overview

Our system comprises three main components: receivers, a wait time estimator, and a database. We show an
out-decision

130 Journal of Signal Processing, Vol. 21, No. 4, July 2017

The wait time estimator detects the overall maximum and new log data as the overall maximum RSSI value. Maximum RSSI value in all the log data including the old and new log data sent from the receiver. We define the all the old log data saved in the database, the other is the

The wait time estimator uses two types of log data. One is the receiver at Point A, where we measure the wait time until a person arrives at the head of a line, and the other receiver at Point B at the head of a line. Smartphones broadcast BLE advertisement packets and the receivers receive them. The receivers record two types of data: the recorded time and the RSSI values. The wait time estimator estimates the wait time using the recorded log data at the two receivers. Then our system provides the estimated wait time through the website to the queueing users.

3.2 Configuration

3.2.1 Transmitter and receiver

In this study, smartphones act as the broadcaster and the receivers act as the observer. A barrier wall is installed in receivers, which was applied in our previous proposed method [2]. All barrier walls are set in the direction in which the users walk. The barrier walls block direct waves while the user is behind the receiver, and the user’s body blocks them while the user is in front of it.

The receivers record two types of data: the recorded time and the RSSI values. Then, the receivers send the recorded log data to the wait time estimator at a certain interval.

3.2.2 Wait time estimator

The wait time estimator uses the wait time estimation algorithm which includes three main subroutines: the maximum RSSI decision, in-decision, and out-decision on the receivers for each user’s smartphone. Figure 2 shows the simplified flow of the wait time estimation algorithm. The wait time estimator first applies data calibration to the recorded log data, which aims to smooth high-frequency noise in the raw RSSI trace. In the data calibration, the raw RSSI trace is filtered by the unique IDs of the advertisement packet and then averaged over one second.

Maximum RSSI decision: The maximum RSSI value is used to recognize whether a user is beside the receiver. The wait time estimator uses two types of log data. One is all the old log data saved in the database, the other is the new log data sent from the receiver. We define the maximum RSSI value in all the log data including the old and new log data as the overall maximum RSSI value. The wait time estimator detects the overall maximum

RSSI value whenever receiving new log data. This overall maximum RSSI value is used in the out-decision.

In-decision: The RSSI value is basically related to the distance. When a receiver continuously obtains a high RSSI value for a certain period of time, the wait time estimator decides that the user is approaching the receivers. We call this the in-decision. We define the in-decision threshold based on a certain period of time and a high RSSI value as the proximity period and the in-border value of RSSI, respectively. To set the in-decision threshold, we conducted a preliminary experiment as described in Sect. 4.2.

Out-decision: Next, when the RSSI value markedly decreases from a higher value, the wait time estimator decides that the user is moving past this receiver point. We call this the out-decision. The wait time estimator uses filtered log data consisting of the log data between a slightly lower value than the overall maximum RSSI value and the overall maximum RSSI value. We define the out-decision threshold based on a decrease in the value of the RSSI as the range value [2].

We show an example of the measured RSSI while a user walks past two receivers in Fig. 3. The solid line and dashed line respectively show the measured RSSI fluctuations at Points A and B. When the wait time estimator determines the in- and out-decisions as true, the wait time estimator sets time stamps $t_{a_{in}}$ and $t_{b_{out}}$ for the earliest received time and the latest received time in the filtered log data, respectively. In particular, the wait time estimator sets time stamps $t_{A_{in}}, t_{A_{out}}, t_{B_{in}},$ and $t_{B_{out}}$ when the user enters and leaves Points A and B, respectively. The estimated wait time is the difference $t_{B_{out}} - t_{A_{in}}$.

3.2.3 Database and website

The wait time estimator stores the user information $t_{A_{in}}, t_{A_{out}}, t_{B_{in}},$ and $t_{B_{out}}$ and the estimated wait time for each user in the database. Also the estimated mean wait time and log data are stored in the database. The website requests the estimated mean wait time and receives it from the database. The website shows the latest estimated mean

Figure 2: Simplified algorithm of wait time estimator

Figure 3: Example of the RSSI at Points A and B

| RSSI (dBm) | In-border value | Overall maximum RSSI |
|-----------|----------------|----------------------|
| Raw RSSI  | In-decision    | Out-decision         |
|          | In-border period | Overall maximum RSSI |

Measurement Point A Measurement Point B

A-in, B-in, A-out, B-out, and A-out when the user enters and leaves Points A and B, respectively. The estimated wait time is the difference $t_{B_{out}} - t_{A_{in}}$.

| Time (s) | A-in, B-in, A-out, B-out, A-out |
|----------|---------------------------------|
|          | Estimated wait-time             |
|          |                                 |

3.2.3 Database and website

The wait time estimator stores the user information $t_{A_{in}}, t_{A_{out}}, t_{B_{in}},$ and $t_{B_{out}}$ and the estimated wait time for each user in the database. Also the estimated mean wait time and log data are stored in the database. The website requests the estimated mean wait time and receives it from the database. The website shows the latest estimated mean

| RSSI (dBm) | In-border value | Overall maximum RSSI |
|-----------|----------------|----------------------|
| Raw RSSI  | In-decision    | Out-decision         |
|          | In-border period | Overall maximum RSSI |

Measurement Point A Measurement Point B

A-in, B-in, A-out, B-out, and A-out when the user enters and leaves Points A and B, respectively. The estimated wait time is the difference $t_{B_{out}} - t_{A_{in}}$.

| Time (s) | A-in, B-in, A-out, B-out, A-out |
|----------|---------------------------------|
|          | Estimated wait-time             |
|          |                                 |
3.3 Implementation
We show the system implementation in Fig. 4. The receiver sends log data to the wait time estimator every 15 s. The wait time estimator estimates the wait time and stores the estimated wait time and the log data in the database every 30 s. The website requests the estimated mean wait time in a fixed interval. The website shows the latest estimated mean wait time, obtained by averaging the latest three estimated wait times.

4. Experiments and Results
We used the iOS devices of the people cooperating in the experiment as BLE transmitters (iPod touch (6th generation), iPhone 6, and 6S). These devices broadcast iBeacon from iOS applications. In addition, the Wi-Fi was turned off in these devices because BLE and Wi-Fi use the same 2.4 GHz frequency band. Advertisement packets always have the same length and are composed of a series of fixed fields. The advertisement interval is preset in the iOS application and we cannot change it. We set various values on Minor, where Minor is a unique ID, to distinguish cooperating users.

We used two Raspberry Pi devices (model B+) equipped with a BLE dongle as the BLE wave receivers. The two receivers and a laptop PC were connected so that their times were synchronized. The receivers record only the iBeacon frame with the same as UUID and Major value in our setting. The recorded items in the receivers consisted of the Minor, the RSSI, and the recorded time.

4.1 Experimental environment
All experiments were conducted in a passageway with eight standing points in a 7 m line. We show the experimental environment in Fig. 5. All users were asked to hold a device in their hands. The 1 m and 6 m points corresponded to Points A and B, respectively. At these points, a stand with a mounted receiver was placed. In our previous study, the estimated wait time was obtained with higher accuracy when the range value was set to 4-8 dB [2]. Therefore we set this value to 6 dB in this study.

4.2 Preliminary experiment
In the preliminary experiment to set the proximity period and the in-border value, we measured RSSI values from the transmitter for 5 min when the one user was standing beside the receiver point.

Figure 6 shows the measured RSSI values. The probability of the measured RSSI value being over -50 dBm was extremely high. Although an RSSI value below -50 dBm was observed from an iPod touch, this was for only 1 s in the 5 min experiment. Therefore, we set the proximity period to 5 s and the in-border value to -50 dBm. Note that our parameters depend on the experimental environment and the type of smartphone.

4.3 Experimental evaluation
In the initial condition, no user was in the line. The users lined up one by one every 10 s, as shown in Fig. 7. When users were standing at all eight standing points, all users waited for 1 or 2 min, and then moved forward one position, as shown in Figs. 8 and 9. The cycle of waiting times is expressed as \{2, 1, 1, 2, 2, 1, 1, 2\} in minutes. Therefore, as shown in Table 1, the expected wait time is different for each user.

This experiment used nine smartphones: three iPod touch, two iPhone 6, and four iPhone 6S devices. Each smartphone, BLE was turned on when the user reached the starting point (0 m) and turned off when the user left ending point (7 m). We performed the experiment two times and obtained the total of 18 measurements.
We show the results of the experiment in Table 1. Non-estimated items are expressed as zero. The expected wait time is different for each user. For example, the user holding device number 5 took 0 s to reach 3 m, which is the first standby point after passing point A. After that, this user waited there a total of 30 s corresponding to a wait for every 10 s for each of the remaining three users to line up. Since this user repeatedly waited 1 or 2 min then move, waited a total of 6 min and moved a total of four times before passing Point B. Therefore, the expected wait time of this user was 6 min 30 s.

The user holding device number 1 instantaneously passed Points A and B. He did not stand beside Points A and B for more than the proximity period. Also, the users holding device numbers 2-6 instantaneously passed Point A. Thus, the wait time estimator determined the in-decision or out-decision as false, so that the non-estimated items of this device numbers was zero. On the other hand, the users holding device numbers 7-18 stood and waited beside Points A and B. Thus, the wait time estimator determined the in-decision and out-decision as true and then the estimated wait times. The difference between the estimated wait time and the expected wait time was within 10 s for all measurements.

5. Conclusion

In this paper, we have proposed an immediate cooperative line wait time estimation system using BLE on a smartphone. In accordance with the preliminary experiment, we set the proximity period to 5 s and the in-border value to -50 dBm as the in-decision thresholds. Also, considering our previous study [2], we set the range value to -6 dBm as the out-decision threshold. Employing these thresholds, we conducted an experimental evaluation. The difference between the estimated wait time and the expected wait time was within 10 s for all measurements. Note that the allowable difference between them is related to the ratio between the difference and the expected wait time and depends on the user’s requirements.

Acknowledgment

This work was supported by KAKENHI (26730052).

References

[1] M. B. Houston, L. A. Bettencourt and S. Wenger: The relationship between waiting in a service queue and evaluations of service quality: A field theory perspective, Psychology & Marketing, Vol. 15, No. 8, pp. 735-753, 1999.
[2] Y. Miyagawa and M. Ogawa: Cooperative line wait time estimation using BLE on smartphone, Journal of Signal Processing, Vol. 20, No. 6, pp. 267-274, 2016.
[3] R. Cope III, R. Cope and H. Davis: Disney’s virtual queues: A strategic opportunity to co-brand services?, Journal of Business & Economics Research, Vol. 6, No. 10, 2008.
[4] M. Bulut, Y. Yilmaz, M. Demirbas, N. Ferhatosmanoglu and H. Ferhatosmanoglu: LineKing: Crowdsourced line wait time estimation using smartphones, Proc. MobiCASE 2012, pp. 205-224, 2012.
[5] Y. Wang, J. Yang, Y. Chen, H. Liu, M. Grutese and R. Martin: Tracking human queues using single-point signal Monitoring, Proc. MobiSys’14, pp. 42-54, 2014.