**Practical Paper**

# A Proposal of Smartphone Beacons in Stay Estimation System Using BLE

Kota Togawa[†], and Katsuhiko Kaji[†]

[†]Graduate School of Business Administration and Computer Science, Aichi Institute of Technology
{b24716bb, kaji}@aitech.ac.jp

***Abstract*** - People spend 88.9% of their day indoors and are mainly active indoors in physically separated spaces such as their own rooms, laboratories, and conference rooms. Therefore, room-level location information, rather than highly accurate location information, is also valuable. We have proposed a stay estimation system that receives signals from BLE beacons carried by each user and estimates the room location using receivers installed in the environment. However, conventional methods using only physical beacons have problems such as battery replacement, time-consuming initial setup, and users moving from room to room without a physical beacon. In this study, we implement and evaluate smartphone beacons that have high tracking performance, do not require initial setup or battery replacement, and consumes little battery power. Smartphones are often carried around at all times, so they are considered to be highly trackable. We implemented an application that automatically sets the content of physical beacon advertisements and continues to advertise BLE signals periodically. In evaluation experiments, the application was found to have high tracking performance and low battery consumption.

***Keywords***: Room-level indoor localization, BLE beacon, Smartphone application

## 1 INTRODUCTION

Since many indoor activities take place within the confines of rooms, room-level indoor location information is valuable. People spend the majority of their time indoors, as reported in the study that people spend 88.9% of their day indoors [1]. Indoors, people are mainly active in physically separated spaces such as their own rooms, laboratories, and conference rooms. Therefore, not only highly accurate location information but also room-level location information is valuable. For example, there are commercial packages available that use room-level location information to manage attendance [2], monitor congestion, and reserve meeting rooms [3].

A method using radio signals is available for room-level location estimation (hereafter referred to as "room-level localization"). Among such methods, those using BLE signals use dedicated devices that transmit BLE signals (hereinafter referred to as "physical beacons") and a receiver that receives BLE signals. The receiver receives BLE advertising packets from nearby physical beacons. The BLE advertising packets contain the MAC address and UUID. This information can determine which physical beacon the receiver received the BLE signal from. Room-level localization using BLE signals is based on this property.

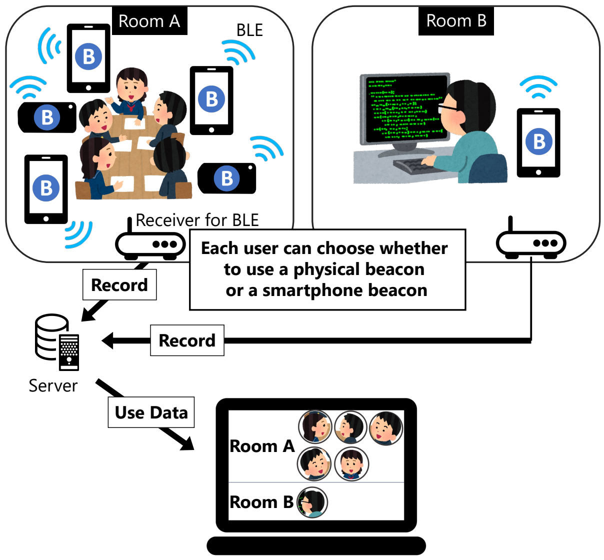



**Figure 1:** Schematic diagram of the stay estimation system to be realized in this study. The user can choose whether to use a smartphone beacon or a physical beacon.

There are two main types of room-level localization methods using BLE signals. The first one is a method in which BLE signals from physical beacons installed in rooms are received by a device such as a smartphone carried by each user and sent to a server. While this method does not require each user to carry exclusive devices, it does require all users to install an application on their own devices. In addition, there is a concern that battery consumption will increase because the application is always running. The second one is to receive BLE signals from the physical beacon carried by each user with a receiver installed in rooms and transmit them to the server. In this method, the users only need to carry the physical beacon distributed by the administrator. We have proposed the system called "StayWatch" that manages the stay information of the members of our laboratory, and we are operating it in our laboratory. The purpose of StayWatch is to activate communication among multiple communities by sharing privacy-conscious stay information.

There are also several issues with the method where the users carry a physical beacon. Since each user carries one physical beacon, the number of beacons needed to be distributed must match the number of users in the community

ISSN1883-4566 © 2026 - Informatics Society and the authors. All rights reserved.

| | Smartphone Beacon | Physical Beacon |
|---|---|---|
| User Burden | Install the application on the user's own smartphone. | Battery needs replacement (About once a year) |
| Administrator Burden | None | Initial setting of UUID, etc. |
| Financial Cost | Low (Use user's smartphone) | High (Thousands of yen per beacon) |
| User Restrictions | No restrictions on app installation | Available to everyone |

**Figure 2:** Features of smartphone beacons and physical beacons

where the beacons are to be deployed. The financial cost is high because many physical beacons are required. In addition, it takes a lot of time and effort to set up physical beacons. When performing room-level localization, individuals are identified using UUIDs advertised from physical beacons. When installing in a community, it is necessary to manually enter and set the UUID of the physical beacon for each user. Other problems include the fact that physical beacons are sometimes placed in rooms and moved from room to room, resulting in low tracking performance, and the fact that some users faling to notice when the batteries in the physical beacons run out, or neglecting to replace the batteries even if they do notice.

The purpose of this research is to realize a BLE beacon with high tracking performance and low financial and human costs. As an approach, we implement and evaluate an application that substitutes the role of a physical beacon with a smartphone (hereinafter referred to as "smartphone beacon"). Smartphone beacons are expected to offer better tracking performance than physical beacons, as many people carry their smartphones with them at all times. An overview of this application is shown in Fig. 1. While smartphone beacons have multiple advantages as shown in Fig. 2, they also have disadvantages such as the need for each user to install the application on their own device. While physical beacons have multiple disadvantages, they also have the advantage that users only need to carry a physical beacon distributed by the administrator, which is less burdensome. In this way, there are advantages and disadvantages for both smartphone beacons and physical beacons. Therefore, the system should be configured so that each user can choose whether to use a smartphone beacon or a physical beacon.

This study makes two contributions. The first contribution is the proposal of a system configuration that enables smartphones and physical beacons to coexist within the same presence estimation system, and the demonstration of its practical feasibility through evaluation experiments. Separate systems existed where users carried either smartphones or physical beacons. However, there was no configuration allowing users to choose whether to carry a smartphone or a physical beacon. The second contribution is the clarification of the advantages of using smartphones as transmitters compared to using them as receivers. Using smartphones as receivers reduces financial costs since no physical beacons need to be prepared for each user, and it also makes third-party tracking more difficult. Therefore, the receiver-based approach has been more widely adopted than the transmitter-based approach. Within this context, this study highlights the advantages of using smartphones as transmitters. Smartphones consume less battery power when transmitting beacon signals than when receiving them. Moreover, because they can operate with the same protocol as physical beacons, coexistence is easier to achieve.

## 2 RELATED RESEARCH

There are various methods for room-level localization using Wi-Fi, BLE, IC cards, cameras, voice recognition, etc. Room-level localization can also be achieved using only information on entry and exit from a room. Methods that take advantage of this are those using IC cards, cameras, and voice recognition. In the method using IC cards, readers are installed at entrances and exits, and users hold their student ID cards or other IC cards over the readers when entering or exiting a room [4]. This requires the user to take active action, which causes operation forgetting. There are methods that do not require active action, such as the voice recognition method where the user says his or her name when entering or exiting a room [5]. Another commercial package is available that uses facial recognition by cameras installed at entrances and exits [6]. However, these methods face issues with reduced accuracy due to users with similar names or poses, as well as the angle of the face captured by the camera.

There are room-level localization methods that use realtime radio signal information, such as Wi-Fi and BLE, as well as entry/exit determination. Some Wi-Fi-based methods detect and use packets sent by smartphones to locate Wi-Fi base stations in the surrounding area [7]. Users do not need to install special applications on their device. However, in recent years, MAC addresses of smartphones are often randomized to improve privacy, in which case this method cannot be used. The fingerprinting method is a method that enables highly accurate location estimation using radio waves [8]. The fingerprinting method has high administrative costs. Room-level localization does not require as much accuracy as the fingerprint method. Therefore, the proximity method, which estimates location at the area level, is often used for room-level localization.

BLE allows for easier adjustment of the location and number of base stations than Wi-Fi. Wi-Fi base stations are installed to improve the wireless communication environment. Therefore, it is difficult to relocate or increase the number of Wi-Fi base stations solely for the purpose of improving the accuracy of room-level localization. In contrast, physical beacons that transmit BLE signals are installed specifically for room-level localization, making it easier to place them in optimal locations and to adjust the number of beacons.

There are two main methods for room-level localization using proximity to BLE beacons: one is to install a physical beacon in rooms and the other is to have users carry physical beacons. In the method that installs physical beacons in rooms, physical beacons are first installed in each room. Each physical beacon should advertise a unique ID. Users run the application on their smartphones to receive signals from the physical beacons. When a user enters a room, their smartphone receives the signals from the physical beacons. The unique ID contained in the signals from the physical beacon is sent to the server to determine which smartphone is being used and which room the physical beacon is in. In this way, the system can estimate who is in each room. Physical beacons do not require any cables; they can be simply fixed to a wall or ceiling. Users do not need to carry any special equipment. A disadvantage is that it is necessary to install a specific application on the smartphone. In addition, the smartphone must continuously receive BLE signals from the surrounding area and communicate with the server, which places a heavy burden on the smartphone and consumes a large amount of battery power.

In the method where each user carries a physical beacon [9], receivers are first installed in each room. The administrator distributes small physical beacons to users, which are configured to advertise a unique ID. When a user enters a room, the receiver in that room receives signals from the physical beacon the user is carrying. The unique ID contained in the signals from the physical beacon and the unique ID of the receiver are sent to the server. The server checks whose physical beacon it is and which room the receiver is in. In this way, it can estimate who is in which room. Users only need to carry the distributed physical beacon in their wallets, bags or similar items. The disadvantage is that the financial cost is high because physical beacons are required for each user. In addition, the initial configuration of physical beacons is time-consuming because it is done manually for each beacon using a configuration application.

When multiple rooms are assumed, depending on the material and thickness of the walls, BLE signals may penetrate walls. In such cases, comparing the signal strength from receivers or BLE beacons installed in each room enables more accurate room estimation.

There is a study that uses BLE signals emitted from smartphones to estimate the level of congestion in a room [10]. For this purpose, the COVID-19 contact tracing application (hereafter referred to as COCOA) [11], which was provided by the Ministry of Health, Labour and Welfare, is used. Smartphones with COCOA installed transmit BLE signals periodically. This BLE signal is received by a receiver installed in the room. The number of smartphones in the surrounding area is determined based on the information about how many BLE signals the receiver receives from them. In this way, the number of people in the room is estimated. Users only need to install COCOA on their smartphones and do not need to install any additional applications for the stay estimation system. Users who do not wish to install COCOA can be provided with physical beacons to carry instead. However, COCOA cannot identify individuals because the contents of advertised packets are encrypted and updated periodically. Therefore, COCOA cannot be incorporated into a stay estimation system.

The acquired stay information can be used in various situations, contributing to the realization of a smart city [12]. For example, it can be used to estimate the room occupancy rates for indoor disaster rescue activities and power management. Another study has been conducted that promotes laboratory visits by predicting and presenting future stay information based on each laboratory member's past stay data and calendar schedules [13].

# 3 SMARTPHONE BEACONS IN STAY ESTIMATION SYSTEMS

In this chapter, we discuss the requirements specification and implementation of the smartphone beacon proposed in this study. Users choose whether to use a physical beacon or a smartphone beacon and carry the chosen device. Receivers in the room receive BLE signals transmitted from the beacon device and send them to the server to record stay information.

## 3.1 Examination of Requirement Specifications

Based on the background of this research presented in Chapter 1 and related research presented in Chapter 2, the requirements for a smart beacon are as follows.

**(R1)** Each user can choose whether to use a smartphone beacon or a physical beacon

**(R2)** Small battery consumption of the smartphone

**(R3)** Small initial setup effort

**(R4)** Users do not need to open the application except when changing settings

**R1** : In order to allow the user to choose whether to use the smartphone beacon or the physical beacon, the smart phone beacon is made to behave in the same way as the physical beacon. It is appropriate for users to be able to choose between smartphone beacons and physical beacons, as each has its own advantages and disadvantages. Physical beacons periodically advertise packets containing a UUID to identify the user. Smartphone beacons should behave in the same way. This would eliminate the need for major modifications to the existing stay estimation system.

**R2** : In order to suppress battery consumption, the application only transmits BLE signals during normal operation. It is necessary to make it a low battery consuming application, because smartphone beacon operates constantly. Periodic transmission of BLE signals can continue for several months to several years with a coin battery. Thus, transmitting BLE signals consumes very little power. Therefore, as in the solution of **R1**, battery consumption can be suppressed by having the smartphone beacon operate in the same way as the physical beacon. The network communication with a smartphone consumes a large amount of battery power. Therefore, the smartphone beacon does not require network communication except when it is set up, and its normal operation is limited to transmitting BLE signals, which consume low battery power.

**R3** : To reduce the time and effort required for initial setup, the application automatically performs the setup. A community administrator performs the initial configuration of the physical beacon using a configuration application and manually enters the UUID and transmission frequency. The initial setup takes more than one minute per beacon, and about one in every 20 beacons is incorrectly configured. Additionally,

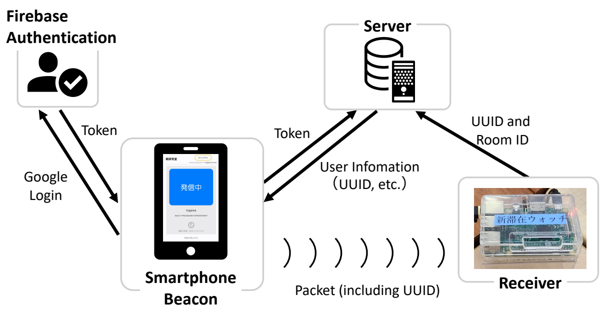



**Figure 3:** Initial setup flow

while the initial setup of physical beacons is done by the administrator, it is not desirable for the administrator to directly operate and configure the smartphone beacon, which is installed on each user's smartphone. We believe that even if the user performs the initial setup, it takes a long time for a person who is familiar with the initial setup to make a mistake, and that an unfamiliar user will take more time and make a mistake. The smartphone beacon automatically performs the initial setup, to reduce this burden on users.

**R4**：To prevent users from forgetting to operate the application, they should not need to open it except during initial setup. Applications that need to be opened periodically require active user action. In the case of such an application, the application may not run due to forgotten operations or troublesome operations. Therefore, the user does not need to actively open the smartphone beacon except for the initial setup or when changing settings. Therefore, smartphone beacons should continue operating unless users intentionally stop it.

## 3.2 How to Set Up Outgoing Packets

Figure 3 shows the flow of automated initial setup to satisfy **R3**. The beacon uses OAuth authentication with a Google account to get the user's UUID. OAuth authentication prevents unauthorized acquisition of the user's UUID. At the first startup, the user logs in with their Google account according to the application screen. A token is issued upon successful authentication with Firebase Authentication. The smartphone beacon sends the token to the server that manages the user information. The server uses the information in the token to verify whether the user has authenticated with Firebase Authentication and whether the user is registered. If the server confirms that the token has been authenticated and the user has been registered, it sends the user information associated with the e-mail address to the smartphone beacon. The beacons use the user information got from the server to perform the initial setup and advertise a packet containing the UUID. In this way, the smartphone beacon satisfies **R3**.

## 3.3 Implementation in Android

We implemented the Android version of the smartphone beacon to satisfy **R1** and **R4**.

### 3.3.1 How to Transmit BLE Signals

To satisfy **R1**, smartphone beacons for Android advertise the UUID got from the server as it is included in the packet. Smartphone beacons transmit BLE signals using android.bluetooth.le package provided by Android. The smartphone beacons use this to set the UUID to be advertised, the frequency of transmission, and the transmission strength, and transmit BLE signals. However, the transmission frequency and strength cannot be strictly set, with only three levels of transmission frequency and four levels of transmission strength available. When the application is not open (hereafter referred to as "background"), the device transmits at the lowest frequency (ADVERTISE_MODE_LOW_POWER) for any of the settings. In order to accommodate large rooms, the transmission strength is set to the strongest setting ADVERTISE_TX_POWER_HIGH. In this way, the smartphone beacon satisfies **R1**.

To satisfy **R4**, the smartphone beacon must also run in the background. For this purpose, the smartphone beacons use a foreground service that allows applications to run in the background. This allows BLE signals to continue being transmitted even if the beacon is task-killed or the screen is put to sleep. However, it is not possible to satisfy **R4**, because this alone may stop the transmission of BLE signals.

### 3.3.2 How to Prevent BLE Radio Transmissions from Stopping

To satisfy **R4**, when the smartphone is unable to continue transmitting BLE signals, it will automatically resume when it is ready to transmit. There are three scenarios where BLE signals transmission in Android becomes uncontrollable: when the Bluetooth function is disabled, when the smartphone is turned off and the foreground service is terminated, or when the foreground service is forcibly terminated by Android.

When the smartphone becomes unable to use the Bluetooth function, the smartphone automatically resumes transmitting BLE signals when it becomes able to do so. When the user turns on Airplane mode, the Bluetooth function becomes unusable and BLE signals transmission stops. When Airplane mode is turned off, the smartphone beacon should automatically resume BLE signals wave transmission. For this purpose, we use the BluetoothAdapter.ACTION_STATE_CHANGED intent broadcasted by Android when the Bluetooth status of the smartphone is changed. When the foreground service detects this intent, it executes a process to start transmitting BLE signals.

When the smartphone is turned off and the foreground service is stopped, BLE signal transmission automatically resumes when the smartphone is turned on. When the smartphone is turned off, the foreground service stops and BLE signal transmission stops. When the smartphone starts up, the foreground service should automatically start and BLE signal transmission should resume. To achieve this, use the ACTION_LOCKED_BOOT_COMPLETED intent broadcast by Android when the smartphone is turned on. When the beacon detects this intent, it starts the foreground service and begins transmitting BLE signals.

When Android forces the application to stop, the BLE transmission of the smartphone beacon is automatically resumed when the resource is reused. If the foreground service has been running for a long time, its priority will gradually decrease, and it may be forced to terminate. In such cases, it is better for the beacon to automatically restart the foreground service and resume BLE signal transmission. To do this, set the return value of the method that is executed when the foreground service is started to START STICKY. When Android kills the foreground service, it will be automatically restarted as soon as the resources become available again. In this way, the smartphone beacon satisfies **R4**.

The smartphone beacon targets devices running Android 7.0 (API level 24) and later. We confirmed its correct operation on devices running Android 11.0(API level 30) through Android 15.0(API level 35).

## 3.4 Implementation in iOS

We implemented the iOS version of the smartphone beacon to satisfy **R1** and **R4**.

### 3.4.1 Restrictions on BLE Signal Transmission in The Background

iOS does not allow a user's UUID to be included in BLE advertisement packets as-is when the application is running in the background. When the application is in the foreground, it is possible to transmit using the iBeacon format, allowing the service UUID to be directly set to the user's UUID, just like on Android or with physical beacons. However, in the background, iOS alters the advertisement data. The iBeacon UUID is placed into an overflow area, resulting in a transformed value being included in the packet.

The value that changes according to the iBeacon UUID in the packet actually advertised by the smartphone beacon is a 152-bit hexadecimal number. Of the 152 bits, 24 bits are fixed values and 128 bits are values that change according to the set UUID. The initial value of the value that changes according to the iBeacon UUID is 0 for all 128 bits. Due to this transformation, the receiver cannot directly retrieve the iBeacon UUID from the packet. Therefore, we configure the iBeacon UUID such that the resulting transformed value can still represent the user's UUID, even after iOS modifies the data in the background.

When a single iBeacon UUID is set, one bit out of 128 bits changes from 0 to 1. As an example, in Fig. 4, the iBeacon UUID is 0466cb00-a9e0-4414-945c-43dd4eec4c6c. In this case, the 19th bit out of the 128 bits changes. The smartphone beacons advertise a value of 4c0001 for the fixed value portion and 00000000000000000000000000080000 for the value that changes according to the iBeacon UUID. Similarly, when the iBeacon UUID 64a84cda-101f-43af-b871-259d37d4309a is set, the 58th bit out of the 128 bits changes. Therefore, the smartphone beacons advertise a value of 4c0001 for the fixed value portion and 0000000000000000040000000 0000000 for the value that changes according to the iBeacon UUID.

When multiple iBeacon UUIDs are set, multiple bits within the 128 bits change from 0 to 1. As an example, the two iBeacon UUIDs given in the previous example are set in Fig. 5. In this case, the 19th and 58th bits out of the 128 bits are changed. he smartphone beacon advertises a value of 4c0001 for the fixed value portion and 0000000000000000040000000 0080000 for the value that changes according to the UUID. When multiple iBeacon UUIDs are set in this way, all the bit positions corresponding to each iBeacon UUID will change.

### 3.4.2 How to Transmit BLE Signals Using The Correspondence Table

We created a correspondence table to express the value to be advertised in the packet using the above properties. The table is shown in Fig. 6.

The smartphone beacon uses the table to set multiple iBeacon UUIDs to represent the values it wants advertised in the packet. The example is shown in Fig. 7. The smartphone beacon wants to advertise 0000000-00a0-0000-0000-0000-00000 00000005000. In this case, the value advertised should be 4c0001 for the fixed value part and 0000000000000000a0000 00000000000000000000000005000 for the value that changes according to the iBeacon UUID. To achieve this, because bits 12, 14, 84, 85, and 87 must be changed from 0 to 1, the iBeacon UUID corresponding to each bit position should be checked in the corresponding table. When all the corresponding iBeacon UUIDs are set and advertised, the smartphone beacon advertises the following value.
4c00010000000000000000000000000000a000000000000 000000000000000000000000000000000000000005000
This allows the UUID to be advertised from an iOS device like a physical beacon.

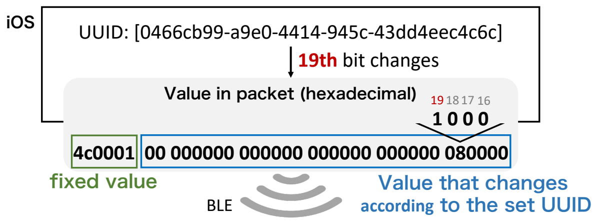



**Figure 4:** Example of transmitting BLE signals in the background on iOS

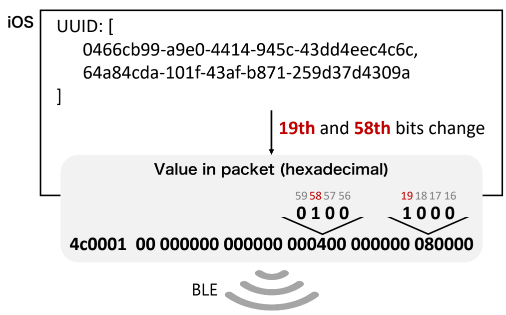



**Figure 5:** Example of multiple UUIDs being sent out in the background on iOS

| Changing bit positions | UUID to be set |
|---|---|
| 0 | 00000000-0000-0000-0000-000000000039 |
| 1 | 00000000-0000-0000-0000-000000000072 |
| 2 | 00000000-0000-0000-0000-00000000002b |
| 126 | 00000000-0000-0000-0000-000000000037 |
| 127 | 00000000-0000-0000-0000-00000000007c |

**Figure 6:** Table of correspondence between bit positions to be changed and iBeacon UUIDs to be set

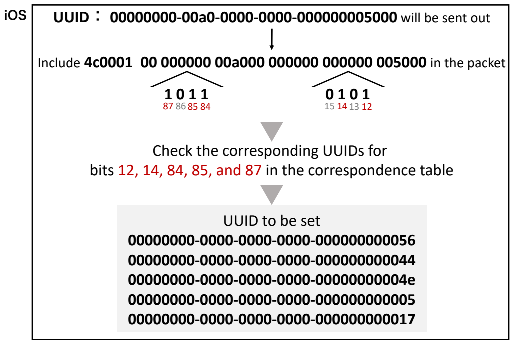



**Figure 7:** Example of setting multiple iBeacon UUIDs with correspondence table in iOS

However, even if the iBeacon UUID to be advertised is expressed in this way, it will conflict with other applications. The overflow area that contains the set iBeacon UUID is common to all applications. Therefore, the value in the packet may be affected by other applications that transmit BLE signals. Even if the user's UUID is contained in the packet, it may be changed by other applications. In this case, the user cannot be identified by the stay estimation system that identifies the user based on a perfect match of UUIDs.

To address this issue, a part of the user's UUID is expressed in multiple places in the packet. The administrator sets the UUID of the iOS user in advance to a UUID that can be identified if a part of the UUID is known. This allows the system to identify the user even if some bits are changed by other applications, as long as other parts are unaffected. In this way, the smartphone beacon satisfies **R1**.

#### 3.4.3 How to Prevent BLE Radio Advertising from Stopping

We made the smartphone beacon automatically resume transmission when it becomes possible to transmit when it is unable to continue transmitting BLE signals to satisfy **R4**. There are two scenarios where BLE signal transmission cannot be continued in iOS: when the smartphone is turned off and when the application is forced to terminate by the system.

We used the state preservation and restoration feature to address these issues. The feature allows the system to take over the background process and start the application in the background when necessary, even if the application is stopped. This allows the system to request that BLE-related tasks be performed in place of the application when it is stopped. In this way, the smartphone beacon satisfies **R4**.

The smartphone beacon targets devices running iOS 6 or later because both CoreBluetooth and state preservation and restoration were introduced in iOS 6. We confirmed its correct operation on devices running iOS 17.0.3 through iOS 18.5.

## 4 EVALUATION EXPERIMENT

### 4.1 Tracking Comparison Experiment

We deployed the smartphone beacon on the stay estimation system to investigate whether the smartphone beacon has better tracking performance than a physical beacon. We conducted an experiment to compare the actual number of stays with those detected by smartphone beacons and those detected by physical beacons.

#### 4.1.1 Experimental Setup

The subjects were 13 university students belonging to the same laboratory. The OS used was Android for subjects 1-6 and iOS for subjects 7-13. The experimental period was from October 10, 2023 to December 26, 2023. However, for iOS subjects, the application built directly from a PC had a validity period of only one week, after which the application would no longer function. Although the application was periodically rebuilt and redistributed, some subjects experienced expiration before reinstallation. Therefore, the experimental period for iOS subjects was limited to the application's validity period. Subjects will be asked to install the smartphone beacon on their own smartphones, and both the smartphone beacon and the physical beacon were kept running during the experiment period. Figure 8 shows an image of the room used for the experiment. One receiver was placed on a desk in the center of the room. A camera was installed at an angle at the room's entrance and exit that allows identification of people entering and exiting. The camera recorded continuously during the experiment period to collect actual stay information. The correct answer rate was defined as the percentage of correctly detected stays out of the actual number of stays.

#### 4.1.2 Experimental Results

Table 1 shows the results of the evaluation experiment. In the table, the correct answer rate for the smartphone beacons is defined as CAR_S and that for the physical beacons as CAR_P. The average correct answer rate for each user was 85% for the smartphone beacon and 62% for the physical beacon. Eleven out of thirteen subjects had a higher correct answer rate for the smartphone beacon. This result suggests that the smartphone beacon has a higher tracking performance than the physical beacon. This may be due to the fact that some subjects left the physical beacon in their room moving between rooms for classes or club activities, whereas many subjects tend to carry their smartphone even when moving between rooms. While some subjects using smartphone beacon

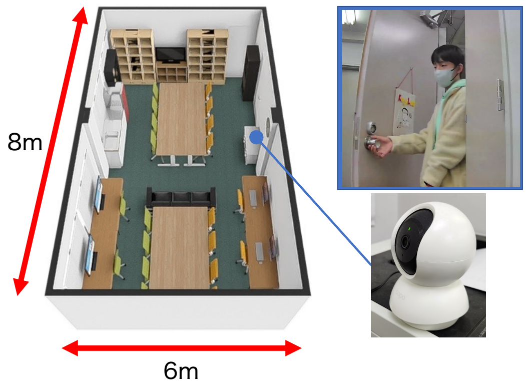



**Figure 8:** Room used for experiments. A camera records the entrance and exit at all times.

**Table 1:** Comparison of tracking performance by subject

| Subject | Number of stays (times) | CAR_S (%) | CAR_P (%) |
|---|---|---|---|
| 1 | 93 | 98 | 92 |
| 2 | 50 | 96 | 92 |
| 3 | 36 | 100 | 36 |
| 4 | 32 | 75 | 78 |
| 5 | 44 | 89 | 82 |
| 6 | 40 | 83 | 38 |
| 7 | 98 | 79 | 60 |
| 8 | 75 | 97 | 32 |
| 9 | 48 | 90 | 54 |
| 10 | 24 | 75 | 50 |
| 11 | 58 | 100 | 50 |
| 12 | 24 | 88 | 63 |
| 13 | 20 | 80 | 90 |
| **avg.** | – | **88** | **63** |

had a correct answer rate of over 90%, others had a correct answer rate of less than 80%.

In order to prevent unexpected behavior in Android, the application stops advertising BLE signals when Bluetooth is turned off. Some subjects frequently turned off the Bluetooth function of their smartphones in order to connect their Bluetooth devices to other devices instead of their smartphones. This was the main reason why the correct answer rate decreased in Android. However, Android can continue to transmit BLE signals even when Bluetooth is turned off. Therefore, it is expected that modifying the implementation will lead to a higher correct answer rate for the smartphone beacon.

## 4.2 Battery Consumption Comparison Experiment

The smartphone beacon suppresses battery consumption, and we conducted an experiment to compare the battery consumption to evaluate whether the beacon was able to satisfy

**Table 2:** 24-hour battery consumption per device

| Device （ad interval, receive interval） | None | Beacon | Scan |
|---|---|---|---|
| Galaxy S10 （130～140ms, 3s） | 8% | 9% | 42% |
| Pixel 7 （130～140ms, 10s） | 6% | 6% | 18% |
| Xperia 10 V （130～140ms, 10s） | 2% | 2% | 5% |
| iPhone 11 Pro （270～280ms, 1s） | 0% | 1% | 3% |

the **R2** defined in 4.1. In order to compare this method with a method in which a physical beacon is placed in the environment and received by a user's smartphone, we implemented an application that receives surrounding BLE signals (hereafter referred to as "scanning application").

### 4.2.1 Experiment Setup

The four smartphones used in the experiment are Galaxy S10, Pixel 7, Xperia 10 V, and iPhone 11 Pro. They are left for 24 hours in each of the following conditions: no activity, with the smartphone beacon running, and with the scanning application running. Each smartphone is charged to 100% beforehand. To prevent external factors, no network connection is made. After 24 hours, we measured the battery consumption. The transmission and reception intervals of the BLE signals are not strictly set by the developer. The transmission interval of the smartphone beacon is 130-140 ms for Galaxy S10, Pixel 7, and Xperia 10 V, and 270-280 ms for iPhone 11 Pro. The receiving interval of the scanning application is 3s for the Galaxy S10, 10s for the Pixel 7 and Xperia 10 V, and 1s for the iPhone 11 Pro.

### 4.2.2 Experiment Results

Table 2 shows the results of the evaluation experiments. Although there are differences depending on the device, in all cases, the smartphone beacon that periodically transmits BLE signals consumes less battery power than the scanning application that continuously receives BLE signals. There is almost no difference in battery consumption between running the beacon and having no application running. This is thought to be due to the fact that transmitting BLE signals consumes little power, and that smartphone beacons are single function applications that only advertise BLE signals.

Pixel 7 and Xperia 10 V consume about three times as much battery power when the scanning application is running as when no application is running. The Galaxy S10 consumes about five times as much. This may be due to the fact that the interval between scans is shorter than that of other devices.

The scanning application is intended for applications that use smartphones to receive BLE signals from physical beacons placed in the environment. This method is expected to consume more battery power than the scanning application used in this experiment, since it also requires periodic transmission of the received physical beacon information to the server. Therefore, the smartphone beacon is considered to be an application with low battery consumption.

## 5 CONCLUSION

In this paper, we discuss the issues of using only physical beacons in a stay estimation system using BLE, and implement and evaluate a smartphone beacon that enables smartphones to behave like a physical beacon. The method using only physical beacons has problems such as high financial cost, time and effort for initial setup, and forgetting to replace batteries. We have developed a smartphone beacon with high tracking performance that does not require initial setup or battery replacement. However, because smartphone beacons also have disadvantages, we made it possible for the user to choose which beacon to use. In the evaluation experiment, we actually installed the smartphone beacons in a stay estimation system and investigated the tracking performance and battery consumption of the smartphone beacons. The results showed that smartphone beacons had higher tracking performance and lower battery consumption than physical beacons.

Current the smartphone beacons are always traceable by unintended third parties because the UUIDs included in the BLE advertising packets are fixed and unchanging for each user. The transmitted BLE signals can be received by electronic devices such as BLE-compatible smartphones, and the packets can be easily browsed. Therefore, as long as the user's UUID is known, a third party can determine when that user was near a receiver by installing a receiver or other device without permission. We believe that a solution to this problem requires a mechanism that allows the stay estimation system to identify whose BLE beacon terminal the signal came from, but prevents third parties from determining the owner of the BLE beacon terminal.

Currently, we only use received radio wave strength for estimation, which may cause misjudgments when the user is in an adjacent room, depending on the material and thickness of the wall. If receivers are installed in adjacent rooms, it would be possible to compare the reception strength of each receiver for more accurate room-level localization. Therefore, it is necessary to deploy and validate our stay estimation system in other communities.

## REFERENCES

[1] C. J. Matz, D. M. Stieb, K. Davis, M. Egyed, A. Rose, B. Chou, and O. Brion, "Effects of age, season, gender and urban-rural status on time-activity: Canadian-Human Activity Pattern Survey 2 (CHAPS 2)." *International journal of environmental research and public health*, Vol. 11, No. 2, pp. 2108-2124 (2014).

[2] ACCESS CO., LTD. Linkit Kintai, https://linkit.access-company.com/kintai/ (reference 2024-11-14).

[3] UCHIDA YOKO CO., LTD. SmartOffice-Navigator, https://www.uchida.co.jp/it/products/smartofficenavigator/ (reference 2024-11-14).

[4] K. Suda, T. Koinuma, and T. Suzuki, "Attendance Registration System Using Contactless IC Cards." *IEICE Technical Report*, Vol. 111, No. 141, pp. 65-70 (2011).

[5] K. Ishizawa, and M. Iwai, "A System of Managing Person Staying in Room Based on Utterance and Recognition Area." *UBI Technical Report*, Vol. 57, No. 19, pp. 1-6 (2018).

[6] Bitkey Inc. workhub, https://www.bitlock.workhub.site/product/face-recognition. (reference 2024-11-14).

[7] A.B.M Musa, and J. Eriksson, "Tracking Unmodified Smartphones Using Wi-fi Monitors." *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pp. 281-294 (2012).

[8] R. Faragher, and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons." *IEEE Journal on Selected Areas in Communications*, Vol. 33, No. 11, pp. 2418-2428 (2015).

[9] P. Barsocchi, A. Crivello, M. Girolami, F. Mavilia, and F. Palumbo, "Occupancy detection by multi-power bluetooth low energy beaconing." *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1-6 (2017).

[10] A. Tokuda, Y. Arakawa, S. Takano, and S. Ishida. "Examination of Automatic Parameter Adjustment in Hybrid Congestion Measurement by WiFi and BLE." *DPS Technical Report*, Vol. 187, No. 16, pp. 1-8 (2021).

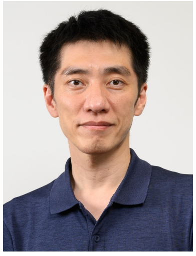
[11] Labour Ministry of Health and Welfare. COVID-19 Contact-Confirming Application (COCOA), (2020) https://www.mhlw.go.jp/stf/seisakunitsuite/bunya/cocoa 00138.html, (reference 2024-11-14).

[12] Cabinet Office. Smart City, https://www8.cao.go.jp/cstp/society5.0/smartcity/, (reference 2024-11-14).

[13] Y. Tanaka, T. Fukushima, and T. Yoshino, "Docoitter：A Presence Display System Capable of Predicting Future In-the-room Information." *IPSJ Journal*, Vol. 54, No. 9, pp. 2265-2275, (2013).

(Received November 15, 2024)
(Accepted August 15, 2025)

**Kota Togawa** is a graduate student at Aichi Institute of Technology. Currently, he is a graduate student of Business Administration and Computer Science, Aichi Institute of Technology. His research interests include indoor positioning.



**Katsuhiko Kaji** received his Ph.D.in information science from Nagoya University in 2007. He be-came a RA at NTT Communication Science Lab-oratories in 2007 and an assistant professor in Nagoya University in 2010. He moved to Aichi Institute of Technology in 2015 as an associate professor, be-coming a professor in 2024. His research interests include indoor positioning, human activity recog-nition, and human augmentation. He is a member of IPSJ.