

Research Article

FloorLoc-SL: Floor Localization System with Fingerprint Self-Learning Mechanism

Kornkanok Khaoampai, Kulit Na Nakorn, and Kultida Rojviboonchai

Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330, Thailand

Correspondence should be addressed to Kultida Rojviboonchai; kultida.r@chula.ac.th

Received 18 June 2015; Revised 28 September 2015; Accepted 30 September 2015

Academic Editor: Paolo Barsocchi

Copyright © 2015 Kornkanok Khaoampai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, a mobile phone plays an important role in daily life. There are many applications developed for mobile phones. Location service application is one kind of mobile application that serves location information. GPS receiver is embedded on a mobile phone for localization. However, GPS cannot provide localization service over indoor scenario efficiently. This is because obstacles and structures of building block GPS signal from the satellites. Many indoor localization systems have been proposed but most of them are developed over single-floor scenario only. The dimension of altitudes in localization results will be missed. In this paper, we propose floor localization system. The proposed system does not need any site survey and any support from back-end server. It has a self-learning algorithm for creating fingerprint in each floor. The self-learning algorithm utilizes sensors on the mobile phone for detecting trace of mobile phone user. This algorithm is low computation complexity, which can be operated on any mobile phones. Moreover, the mobile phone can exchange fingerprints with others via virtual ad hoc network instead of learning all floor fingerprints by themselves only. Our proposed floor localization system achieves 87% of accuracy.

1. Introduction

Mobile phones become a necessity of our daily life. Many applications have been developed to enhance the functionality of mobile phones. Location service application is one of the most interesting applications because it has many advantages, for example, providing an online map, a navigation service, and the current location of people who need a special care or emergency help.

Presently, there is a GPS (Global Positioning System) for localization. The mobile phone is generally embedded with a GPS sensor. The sensor receives the GPS signal from at least four GPS satellites and then calculates the current position. However, the GPS has a limitation. It cannot provide the indoor localization service because the GPS signal is prevented by obstruction and structure of the buildings.

Many indoor localization systems have been proposed to locate a mobile device in a building [1–12]. The previously proposed systems apply fingerprint-based approach [1–4, 6, 7, 9–12]. The concept is that the different location in the

building has a different signal characteristic. Many kinds of signals can be utilized but a Wi-Fi signal is the most popular one for indoor localization systems using fingerprint-based approach. This is because Wi-Fi access points are installed in many buildings for Internet service. The system can be deployed without installing new sensors.

However, the existing indoor localization systems are developed for single-floor scenario. These systems cannot identify the current floor level of the mobile phone because of the interference of Wi-Fi signal between adjacent floors. The signal in adjacent floors is similar. The dimension of altitude will be missed in localization result.

Many researchers realized the above-mentioned problem. The floor localization systems have been developed to identify the current floor level of the mobile phone. SkyLoc [13] utilizes GSM fingerprint for determining the current floor of the mobile phone in multifloor building scenario. SkyLoc requires site survey for localization. FTrack [14] is another research about floor localization. The system applies accelerometer to recognize the user traces and analyzes these

traces to determine the current floor of the mobile phone user. FTrack can work without site survey but it still requires back-end server.

The localization system requiring site survey and back-end server has limitations. The localization service cannot be served if it cannot reach the server due to connection problem or server crashing. The system cannot work without prior knowledge of building. In addition, the back-end server and site survey lead to both financial cost and labor cost.

In this paper, we propose a floor localization system on mobile phone platform named FloorLoc-SL. FloorLoc-SL does not require both back-end server and site survey. The floor localization utilizes a Wi-Fi scanning result to determine a current floor level of the mobile phone. The floor localization algorithm in the system is low complexity. The mobile phone can localize itself using its limited hardware resources. To create fingerprint map construction, the system has the fingerprint self-learning mechanism. The mobile phone is able to learn a new floor fingerprint if it detects a new Wi-Fi signal characteristic. This ensures that the system can provide localization service in many buildings although the system does not have prior knowledge about these buildings. Besides, the mobile phone can exchange floor fingerprints with others via virtual ad hoc network. The virtual ad hoc network applies Wi-Fi hotspot on the mobile phone to act as an ad hoc network. It does not need any supporting infrastructure. The dissemination protocol can perform even if the mobile phone cannot access any infrastructure-based network.

The main contributions of this paper are summarized as follows. (1) We propose a floor localization system on the mobile phone. The floor localization system can add up a dimension of altitude for the previously proposed indoor localization systems. (2) The floor localization system that we propose does not require any back-end server and site survey. To the best of our knowledge, FloorLoc-SL is the first floor localization system that does not require supporting from both the back-end server and site survey. (3) We propose human activity classification and floor counting methodology for fingerprint self-learning using embedded sensors on the mobile phone. The accuracy of activity classification is up to 85% and the floor counting methodology achieves 100% of accuracy.

The structure of this paper is organized as follows. The background knowledge and related work are discussed in Section 2. The overview of our proposed system is presented in Section 3. Section 4 describes how to design and develop our proposed system. Section 5 describes and discusses the experiment. The summary of this paper is concluded in Section 6.

2. Background Knowledge and Related Work

Many researchers have proposed indoor localization systems applying several technologies such as Wi-Fi signal [1, 3, 4, 7, 9, 10, 12], acoustic background [2], inertial sensor on mobile phone [15], and magnetic distortion [11]. The indoor

localization system can be classified into two types: model-based approach and fingerprint-based approach. The model-based approach has a mathematical model to calculate a distance between signal emitter and receiver. The model requires information such as received signal strength index (RSSI), time of arrival (ToA), time difference of arrival (TDoA), and angle of arrival (AoA) for calculating the distance. The fingerprint-based approach assumes that each location has a specific signal characteristic. The localization system using this approach necessarily has a fingerprint map for positioning. This approach needs a site survey for fingerprint map construction and a back-end server for mapping received signal from device to fingerprint map in localization process.

There are many proposed indoor localization systems using fingerprint-based approach [1–4, 6, 7, 9–12]. However, some proposed systems are developed for single-floor scenario. They cannot identify the current floor level due to signal interference between adjacent floors as we mentioned in Section 1. The floor localization systems have been proposed to address the above-mentioned issue.

Marques et al. proposed an indoor localization approach to estimate the position of a person or a mobile device in wireless environments [12]. The approach is fingerprint based. It combines similarity functions and majority rules for estimating the position of person or the device in multibuilding and multifloor scenario. The system constructs a radio map from RSSI value of Wi-Fi signal. The positioning algorithm identifies the building, the floor, the room, and position of device. The system achieves 3.35 m of positioning accuracy, 74.1% of room localization accuracy, and 99.5% of floor localization accuracy. However, the system needs a site survey and a back-end system to support the process of radio map construction.

SkyLoc [13] is GSM fingerprint-based floor localization on the mobile phone. The system achieves up to 73% of accuracy to identify floor level and 97% within two floors. The GSM fingerprint map requires training phase. However, the system can work in the previously surveyed buildings only.

FTrack [14] is a floor localization system proposed by Ye et al. It is infrastructure-free localization. The system uses an accelerometer to collect traces of the mobile phone user and analyzes traces to find the travelling time and step counts between any two floors. FTrack achieves 90% of accuracy but it still needs back-end server for analyzing traces of mobile phone users.

In our previous work, we proposed a low complexity floor localization algorithm named FloorLoc [16]. The algorithm utilizes top- N access points from Wi-Fi scanning list to determine the current floor of the mobile phone. FloorLoc with top- N decrement technique achieves 91% of accuracy. Although the mobile phones can localize themselves without any support from back-end server because the algorithm is low complexity, FloorLoc still needs site survey for collecting top- N access points that the mobile phone received their signal in each floor. This paper improved FloorLoc by removing site survey. The crowded sourcing and activity recognition will be applied for fingerprint self-learning mechanism. The

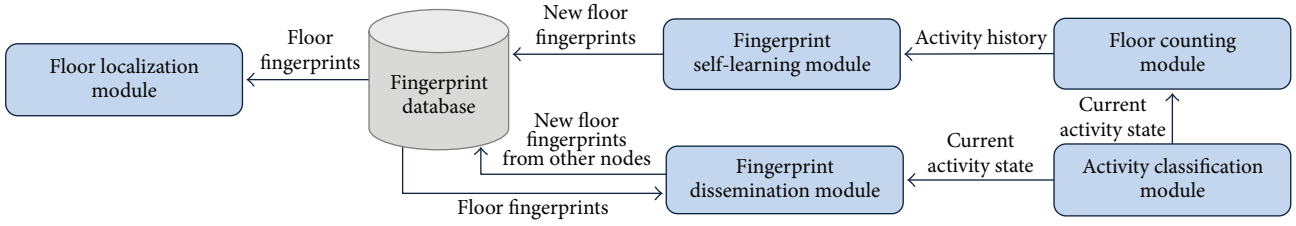


FIGURE 1: System architecture.

proposed system can be used in buildings even if these buildings have not been surveyed.

3. System Overview

FloorLoc-SL was developed as an application on Android platform. There are 5 modules in the system: floor localization module, fingerprint self-learning module, activity classification module, floor counting module, and fingerprint dissemination module. The responsibilities and working relationship of these modules will be described as follows.

Figure 1 illustrates architecture of FloorLoc-SL. The floor localization module has a responsibility to determine the current floor level of a mobile phone using floor localization algorithm named FloorLoc. This module tries to match a received Wi-Fi signal characteristic to floor fingerprint stored in the fingerprint database. The fingerprint self-learning module constructs and updates floor fingerprints to support the localization process of the floor localization module. In the case that the mobile phone cannot localize itself because a received Wi-Fi characteristic does not match with any fingerprint in the database, the fingerprint self-learning module requests activity history from the floor counting module. The activity history is a movement record of the mobile phone user. For example, the mobile phone user has descended downstairs two floors. The activity history is reset every successful localization. The fingerprint self-learning module applies the last known position and the activity history of users to determine the current floor of the mobile phone. The received Wi-Fi characteristic with the current floor of the mobile phone will be added into the fingerprint database as a fingerprint of a particular floor. The floor counting module needs to know all activities of the mobile phone user for counting ascending/descending floors. The activity classification module processes the current activity state of the mobile phone user and serves all activity states for floor counting module. Moreover, the mobile phone does not learn new floor fingerprint by itself; it can exchange its floor fingerprints in the database with others by the fingerprint dissemination module. The operation of these modules will be described in Section 4 for more clarification.

The workflow of FloorLoc-SL is shown in Figure 2. The mobile phone requests a building name and a beginning floor from which the mobile phone user launches the application. The mobile phone adds fingerprint of the beginning floor by fingerprint self-learning module and then localization process is operated. After that, the mobile phone exchanges

its floor fingerprints valid in the database by fingerprint dissemination module. The floor fingerprint has been exchanged until the mobile phone user starts ascending or descending to other floors. The floor counting module will begin to update the activity history. The current activity of the mobile phone user is detected and classified by activity classification module. The localization will be processed again when the mobile phone user stops ascending and descending to other floors. In the case the mobile phone cannot determine its current floor level, it learns a new floor fingerprint by the fingerprint self-learning mechanism.

4. System Design

4.1. Floor Localization Module. The floor localization module is proposed for determining a current floor level of the mobile phone. The localization process in this module is a fingerprint-based approach. The module utilizes a received Wi-Fi signal to determine the current floor level. This module has a floor fingerprint to identify each floor in the building and a floor localization algorithm to match the received Wi-Fi signal and the floor fingerprint.

4.1.1. Floor Fingerprint Characteristic. In FloorLoc-SL, the floor fingerprint is a collection of top- N access points from Wi-Fi scanning list scanned in a particular floor in a building. The Wi-Fi scanning list is a list of access points, whose signal the mobile phone can hear. The access points in the list are ordered by received signal strength (RSS). An access point is identified by its MAC address.

The data structure of floor fingerprint is defined as follows.

$floor_fingerprint = \{floor_level, ap_list\}$, where $floor_level$ is a floor level and ap_list is a list of top- N access points from Wi-Fi scanning list. The ap_list is defined as $\{ap_1, ap_2, \dots, ap_N\}$, where ap_i is a MAC address of each top- N access point from Wi-Fi scanning list.

4.1.2. Floor Localization Algorithm with Top- N Decrement Technique. We utilize the floor localization algorithm named FloorLoc [16] for this system. In order to determine the current location of the mobile phone, the algorithm measures a similarity between top- N access points from Wi-Fi scanning list and floor fingerprints in the database. A variable indicating similarity is a *hit score*. The algorithm assigns the *hit score* for each fingerprint. However, there is the case that many floor fingerprints provide same maximum *hit*

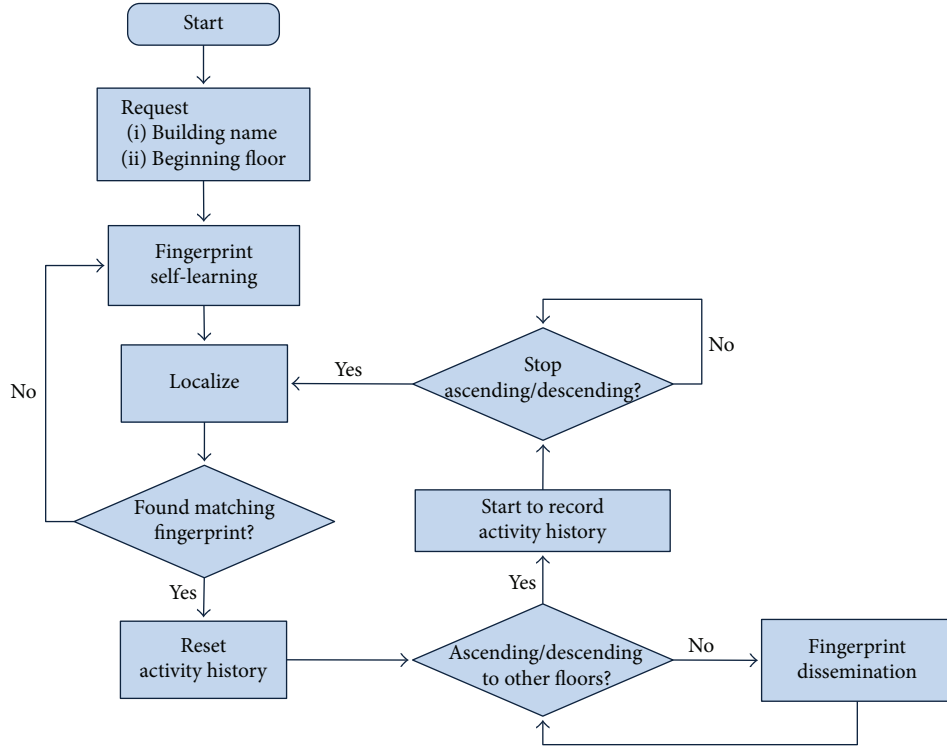


FIGURE 2: System workflow.

score. This situation affects the fact that the algorithm cannot determine the current floor level.

In order to solve the unknown localization problem, the floor localization algorithm applies top- N decrement technique: a technique for reducing unknown localization results. FloorLoc with top- N decrement technique will decrease amount of top- N access points when the maximum *hit score* is provided from many floor fingerprints. The complexity of floor localization algorithm with top- N decrement technique is $O(F)$ where F is the number of floor fingerprints valid in the fingerprint database. The pseudocode of the algorithm is shown in Algorithm 1.

4.2. Fingerprint Self-Learning Module. The task of the fingerprint self-learning module is to help the mobile phone to learn a new floor fingerprint including updating floor fingerprints by itself. The module is operated whenever the mobile phone detects a new Wi-Fi scanning list and the scanning list does not match with any floor fingerprint stored in the mobile phone. This module applies activity history and the last known location to determine the current floor level of the mobile phone. If the mobile phone knows its current location, it can add the Wi-Fi scanning list as a new fingerprint or update fingerprint of this floor in the case this floor fingerprint has been valid in the fingerprint database. The floor fingerprint algorithm is proposed in order to create and update floor fingerprint.

The concept of floor fingerprint algorithm is an accumulation of top- N access points from Wi-Fi scanning list scanned at a particular floor. The scanning lists from different

locations in the same floor do not have to be the same. The mobile phone may discover new top- N access points when it scans Wi-Fi signal in the new position that has never been scanned before. That depends on a distance between a scanning position and access points. The nearest access point is placed as a first rank in the scanning list because it provides the strongest Wi-Fi signal.

The floor fingerprint algorithm is an incremental learning methodology. The fingerprints can be updated when the mobile phone found new top- N access points from scanning in a particular floor. The pseudocode of the algorithm is shown in Algorithm 2. The complexity of the algorithm is $O(N)$ when using top- N access point as a floor fingerprint.

Figure 3 is an example of fingerprint self-learning methodology. In Figure 3(a), the floor localization module cannot match a received Wi-Fi scanning list to any floor fingerprints. However, the fingerprint self-learning module knows an activity history. The value “up, 2 floors” of activity history means the mobile phone user has gone up two floor levels. The module found that the last known location of the mobile phone user is the 1st floor. Therefore, the module will conclude that the 3rd floor is the current location of the mobile phone. The Wi-Fi scanning list will be added as fingerprint of the 3rd floor as shown in Figure 3(b).

4.3. Activity Classification Module. This objective of this module is to provide a current activity state of mobile phone user for floor counting module. Due to the fact that FloorLoc-SL is a localization system on mobile phone platform without any support from back-end server, the activity classification

Input: Top-N access points in Wi-Fi scanning list *top_ap_list*
Output: Current floor of mobile phone *current_floor_level*

```

(1) current_floor_level = UNKNOWN
(2) hitMAX ← 0
(3) while top_ap_list is not empty do
(4)   for each floor_fingerprint do
(5)     hit ← 0
(6)     for each access point from top_ap_list do
(7)       if access point is in ap_list of floor_fingerprint then
(8)         hit ← hit + 1
(9)       end if
(10)    end for
(11)    if hitMAX = hit do
(12)      current_floor_level ← UNKNOWN
(13)    else if hitMAX < hit do
(14)      hitMAX ← hit
(15)      current_floor_level ← floor_level of floor_fingerprint
(16)    end if
(17)  end for
(18)  if current_floor_level = UNKNOWN do
(19)    top_ap_list.remove(last access point of top_ap_list)
(20)  else do
(21)    break
(22)  end if
(23) end while
(24) return current_floor_level

```

ALGORITHM 1: FloorLoc with top-N decrement.

Input: top-N access points in Wi-Fi scanning results *top_ap_list*,
 floor level *floor*

```

(1) if fingerprint of floor is valid in fingerprint database then
(2)   floor_fingerprint ← fingerprint of floor in fingerprint database
(3) else do
(4)   floor_level ← floor
(5)   ap_list ← null
(6)   floor_fingerprint ← floor_level, ap_list
(7) end if
(8) for each access point from top_ap_list do
(9)   if access point is not in ap_list of floor then
(10)    ap_list.add(access point)
(11)   end if
(12) end for

```

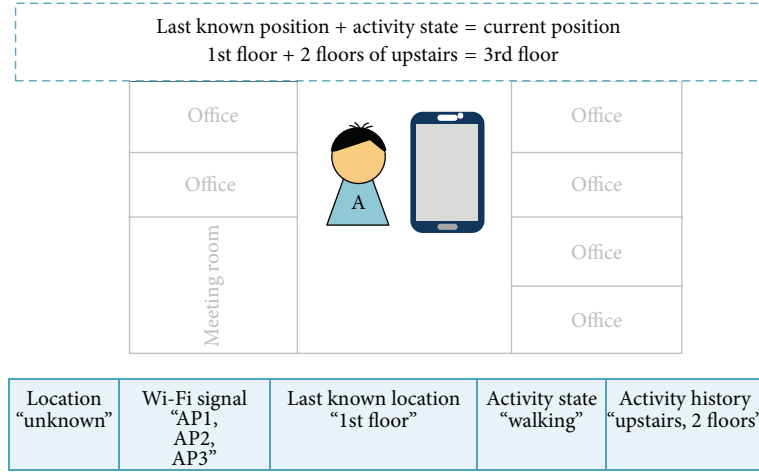
ALGORITHM 2: Fingerprint algorithm.

algorithms we proposed have to be operated with limited resource of mobile phone. That means the algorithm including their parameter tuning must be low complexity and can be processed on the mobile phone.

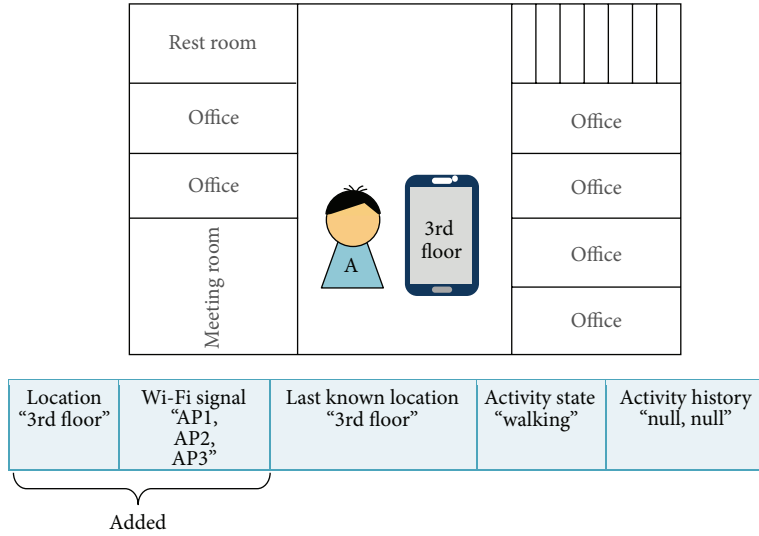
4.3.1. Categories of Human Activity. The human activities are classified into two types: horizontal activity and vertical activity. The horizontal activity is a movement that does not bring about changing the current floor of the mobile phone. The activity state is “unchanged” when the mobile phone user is moving as horizontal activity. The horizontal activity can

be walking, running, and stationary. The vertical activity is an activity, which makes the mobile phone user change the current floor level such as going up the stairs, going down the stairs, or taking elevator. The vertical activity should indicate a movement direction. There are two possible activity states in vertical activity: “up” and “down.” The activity state “up” means the mobile phone user is moving to a higher floor and “down” means the mobile phone is moving to a lower floor.

4.3.2. Activity Classification Approaches. There are many proposed activity classification methodologies for mobile phone



(a) The mobile phone calculates a current floor level



(b) The mobile phone adds new fingerprint of 3rd floor

FIGURE 3: The example of fingerprint-self learning methodology.

platform [17–20]. Most of them are model-based approach. Their methodologies for creating model and tuning parameters are high complexity. The models cannot be created and calibrated on the mobile phone. Due to the fact that FloorLoc-SL is a distributed system without any infrastructure support, the algorithms including their parameter tuning should be processed on the mobile phone. Therefore, our proposed system needs an activity classification algorithm that is low complexity for mobile phone platform.

In FloorLoc-SL, we propose two approaches of the activity classification. One is an acceleration approach. This approach uses acceleration from an accelerometer. Another is an atmospheric pressure approach. This approach applies an atmospheric pressure from barometer sensor to determine the current activity. Both of them are low complexity algorithms. They can be processed with limited resources of the mobile phone.

(i) Acceleration Approach

(a) *Accelerometer.* An accelerometer is a sensor measuring acceleration. Many models of mobile phones are embedded with the accelerometer. The accelerometer reports acceleration in terms of a_x , a_y , and a_z where a_x , a_y , and a_z are acceleration in X, Y, and Z axis, respectively. The magnitude of acceleration, a , can be computed using

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}. \quad (1)$$

(b) *Acceleration Characteristic of Each Activity.* In this approach, we consider stationary and walking activities as an activity state "unchanged" in horizontal activity and we focus on going up the stairs as an activity state "up" and going down the stairs as an activity state "down" in vertical activity. Going to other floors by elevator or escalator will be considered in the future work.

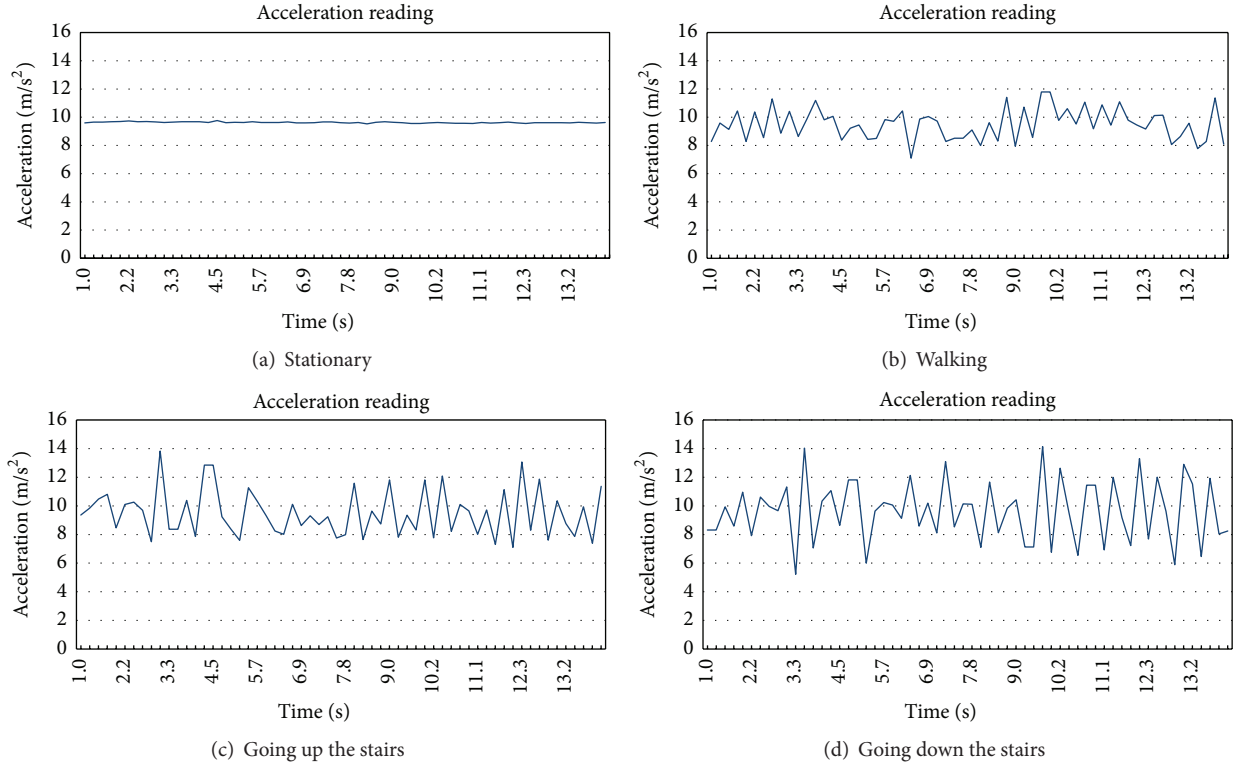


FIGURE 4: The acceleration of each activity.

The graphs in Figures 4(a)–4(d) show acceleration, a , while a user was performing each activity. In each graph of activity, gaps between peak and valley value are different. The gaps are close to zero in the graph of stationary activity (Figure 4(a)). The gaps in the graph of walking (Figure 4(b)) are less than the gap of ascending stairs (Figure 4(c)). The maximum gap is found in the graph of going down the stairs (Figure 4(d)).

(c) Activity Classification Algorithm. The activity classification algorithm is a simple magnitude based. It utilizes a difference between the current and its previous magnitude of acceleration. The algorithm has a set of thresholds to identify the current activity of the mobile phone user: α_{down} for going down the stairs, α_{up} for going up the stairs, and α_{walk} for walking. The current activity is determined by comparing with these thresholds. The optimal value of α_{walk} , α_{up} , and α_{down} will be discussed in Section 5.

However, acceleration reported from accelerometer has noises. For example, we found small gaps in Figure 4(d). The result from activity classification algorithm will be walking even if the actual activity is going down the stairs. In order to address this issue, we have enhanced the activity classification algorithm by applying with majority voting technique. There is a fixed-size window in the activity classification module. The accelerometer feeds measured acceleration to the windows. The algorithm calculates a difference of magnitude, determines an activity, and fills the result in the window. If the window is full, the algorithm

looks for the major activity in the window. The major will be the current activity of the mobile phone user. The algorithm clears the window to prepare for the next classification. Algorithm 3 shows the pseudocode of activity classification algorithm using this acceleration approach. The complexity of algorithm is $O(W)$ where W is a size of window.

(ii) Atmospheric Pressure Approach

(a) Atmospheric Pressure. An atmospheric pressure is a force from the weight of the air acting on a unit of surface in atmosphere of Earth. The atmospheric pressure decreases when an elevation is increased and vice versa. The mobile phone can measure atmospheric pressure by a barometer. Nowadays, many models of mobile phones are embedded with the barometer sensor for more functions. The atmospheric pressure reported by the barometer sensor on the mobile phone is in a format of hPa unit (hectopascal, 10^2 Pa).

(b) Atmospheric Pressure of Each Activity. The graph in Figure 5 shows atmospheric pressures recorded while the mobile phone user was performing horizontal and vertical activity. The pressures were collected from 17th–20th floor of Charoen Visawakarma Building in Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand. This building is a prototype of our proposed system. The timeline shown below the graph represents a sequence of actual activities

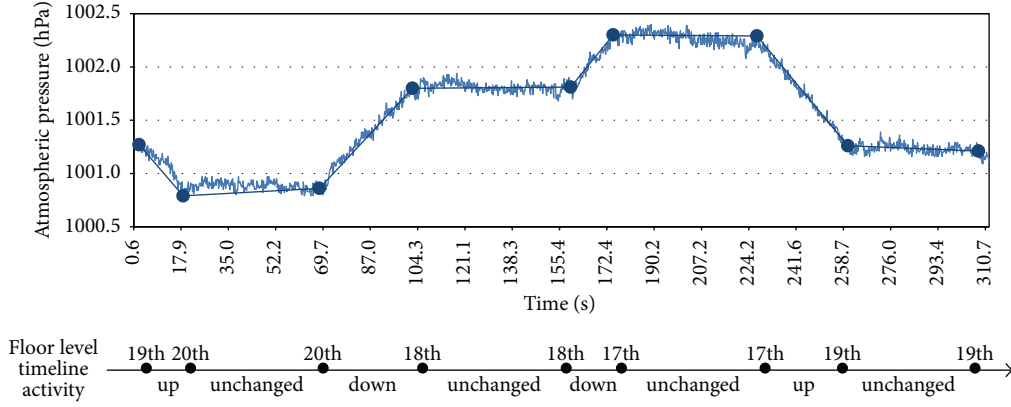


FIGURE 5: The atmospheric pressure from barometer sensor.

Input: acceleration $\{a_1, a_2, a_3, \dots, a_{\text{window size}+1}\}$
Output: current activity *activity_state*

```

(1) window  $\leftarrow \{\}$ 
(2) for  $i = 1$  to  $i = \text{window size}$  do
(3)    $\Delta a \leftarrow |a_i - a_{i+1}|$ 
(4)   if  $\Delta a \geq \alpha_{\text{down}}$  then
(5)     activity  $\leftarrow \text{DOWN}$ 
(6)   else if  $\Delta a \geq \alpha_{\text{up}}$  then
(7)     activity  $\leftarrow \text{UP}$ 
(8)   else if  $\Delta a \geq \alpha_{\text{walk}}$  then
(9)     activity  $\leftarrow \text{WALK}$ 
(10)  else then
(11)    activity  $\leftarrow \text{STEADY}$ 
(12)  end if
(13)  window.add(activity)
(14)   $i \leftarrow i + 1$ 
(15) end for
(16) activity_state  $\leftarrow$  majority activity in window
(17) return activity_state

```

ALGORITHM 3: Activity classification using acceleration approach.

of the mobile phone user. The pressure tends to be stable while the mobile phone user was performing the “unchanged” activity state such as walking or being stable. The pressure becomes lower while the mobile phone user was going up and higher while the mobile phone user was going down to other floors.

(c) *Activity Classification Methodology.* The activity classification algorithm using the atmospheric pressure approach classifies the current activity by expecting decrement and increment of pressure. The pseudocode of the activity classification algorithm using atmospheric approach is shown in Algorithm 4.

The pressures are reported as a stream. The streamed pressures are sliced to be a fixed-size window. The algorithm calculates an average value of the window and then it compares with the average value of the previous window. If the average value of the current window increases more

than β_{down} , the current activity state will be “down” and if the average value decreases more than β_{up} , the current activity state will be “up.” The current activity state will be “unchanged” in the case when the average value does not change more than both thresholds. The optimal value of β_{up} and β_{down} will be evaluated in Section 5.

4.4. Floor Counting Module. The floor counting module has a responsibility to counting floors that the mobile phone user has gone up or down to support fingerprint self-learning module. In this paper, two approaches are proposed for counting amount of floors: acceleration approach and atmospheric pressure approach.

4.4.1. Floor Counting by Acceleration Approach. This approach has a state machine to counting amount of floors. The state machine is an activity-driven based. A current activity state is a condition for state transition. The state machine is shown in Figure 6. The machine is specially designed for counting the floors in a building with one stairs landing between adjacent floors. The number of floors the user goes up/down is valid in *FLOOR_COUNT*. The sign of *FLOOR_COUNT* represents direction of vertical movement. The positive *FLOOR_COUNT* and negative *FLOOR_COUNT* mean the user goes up or down, respectively.

4.4.2. Floor Counting by Atmospheric Pressure Approach. As we mentioned before, the atmospheric pressure is different in each floor level of a building. The atmospheric pressure between adjacent floors is different constantly. The atmospheric pressure of lower floor level is more than that of higher floor level. The floor counting algorithm using the atmospheric pressure approach applies this concept. The pseudocode of the algorithm is shown in Algorithm 5. The complexity of processing each reported pressure in the algorithm is $O(1)$.

In the algorithm, a parameter *initial* represents an atmospheric pressure of the current floor. After initializing this parameter, the algorithm waits until a difference of atmospheric pressure can reach to the thresholds for one-floor

Input: streamed atmospheric pressure $\{p_1, p_2, p_3, \dots\}$
Output: current activity *activity_state*

```

(1) previous_pressure  $\leftarrow \text{average}(p_1, p_2, p_3, \dots, p_{\text{window\_size}})$ 
(2) for  $i \leftarrow \text{window\_size} + 1$  to  $i = n$  do
(3)   current_pressure  $\leftarrow \text{average}(p_i, p_{i+1}, p_{i+2}, \dots, p_{i+\text{window\_size}-1})$ 
(4)    $\Delta p \leftarrow \text{current\_pressure} - \text{previous\_pressure}$ 
(5)   if  $\Delta p \geq \beta_{\text{down}}$  then
(6)     previous_pressure  $\leftarrow \text{current\_pressure}$ 
(7)     activity_state  $\leftarrow \text{DOWN}$ 
(8)   else if  $\Delta p \leq \beta_{\text{up}}$  then
(9)     previous_pressure  $\leftarrow \text{current\_pressure}$ 
(10)    activity_state  $\leftarrow \text{UP}$ 
(11)  else then
(12)    activity_state  $\leftarrow \text{UNCHANGED}$ 
(13)  end if
(14)   $i \leftarrow i + \text{window\_size}$ 
(15) end for

```

ALGORITHM 4: Activity classification using atmospheric pressure approach.

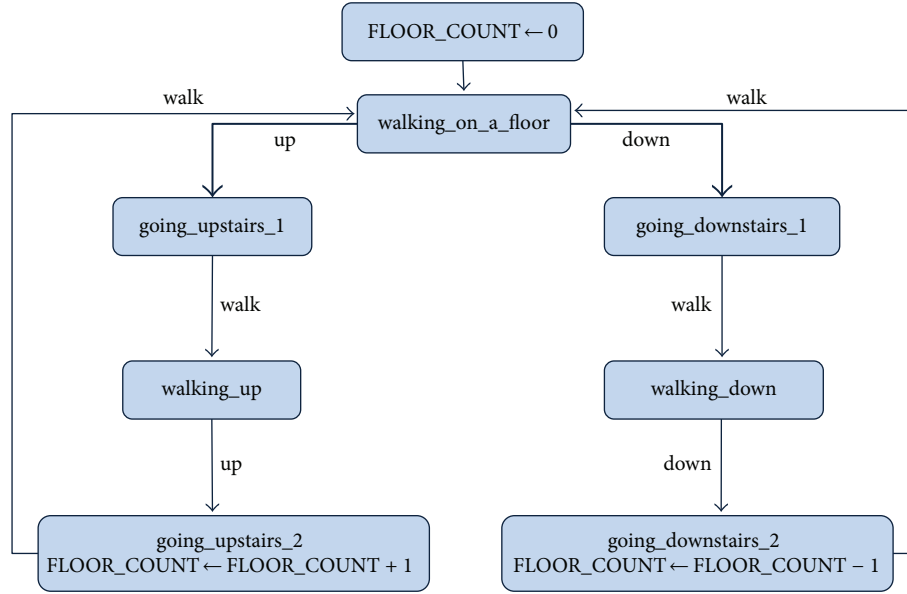


FIGURE 6: State machine for the floor counting module with acceleration approach.

detection ($\beta_{\text{floor_down}}$ and $\beta_{\text{floor_up}}$) and then it will update the parameter *initial* with the atmospheric pressure of the new floor. The algorithm has a parameter Δ to indicate the difference between an atmospheric pressure p_i and the parameter *initial*. If the parameter Δ is more than or equal to $\beta_{\text{floor_down}}$, that means the mobile phone user goes down a floor and if the parameter Δ is less than or equal to $\beta_{\text{floor_up}}$, that means the mobile phone user goes up a floor. The number of floors that the mobile phone user goes up/down is valid in *floor_count*. If *floor_count* is negative, that means the mobile phone user goes down to other floors and if *floor_count* is positive, that means the mobile phone user goes up to other floors. The value for threshold $\beta_{\text{floor_down}}$ and $\beta_{\text{floor_up}}$ will be evaluated in Section 5.

4.5. Fingerprint Dissemination Module. Besides the fingerprint self-learning module, the fingerprint dissemination module is another way to collect new floor fingerprints. Conceptually, all mobile phones can provide floor fingerprints valid in their databases to other nodes. The floor fingerprints are disseminated via a virtual ad hoc network by fingerprint dissemination protocol.

4.5.1. Fingerprint Dissemination Protocol. Figure 7 illustrates steps of fingerprint dissemination. At the beginning, a dissemination node broadcasts a beacon message. Other nodes receive the beacon message. They check the floor fingerprints in their databases and then they request floor fingerprints that are not valid in the database by sending a

Input: streamed atmospheric pressure $\{p_1, p_2, p_3, \dots\}$
Output: amount of floors $floor_count$

```

(1)  $initial \leftarrow p_1$ 
(2)  $floor\_count \leftarrow 0$ 
(3) for each  $p_i$  from streamed atmospheric pressure do
(4)   if  $p_i \neq 0$  then
(5)      $\Delta \leftarrow initial - p_i$ 
(6)     if  $\Delta \geq \beta_{floor\_down}$  then
(7)        $initial \leftarrow p_i$ 
(8)        $floor\_count \leftarrow floor\_count - 1$ 
(9)     else if  $\Delta \leq \beta_{floor\_up}$  then
(10)       $initial \leftarrow p_i$ 
(11)       $floor\_count \leftarrow floor\_count + 1$ 
(12)     end if
(13)   end if
(14) end for

```

ALGORITHM 5: Floor counting algorithm using atmospheric pressure approach.

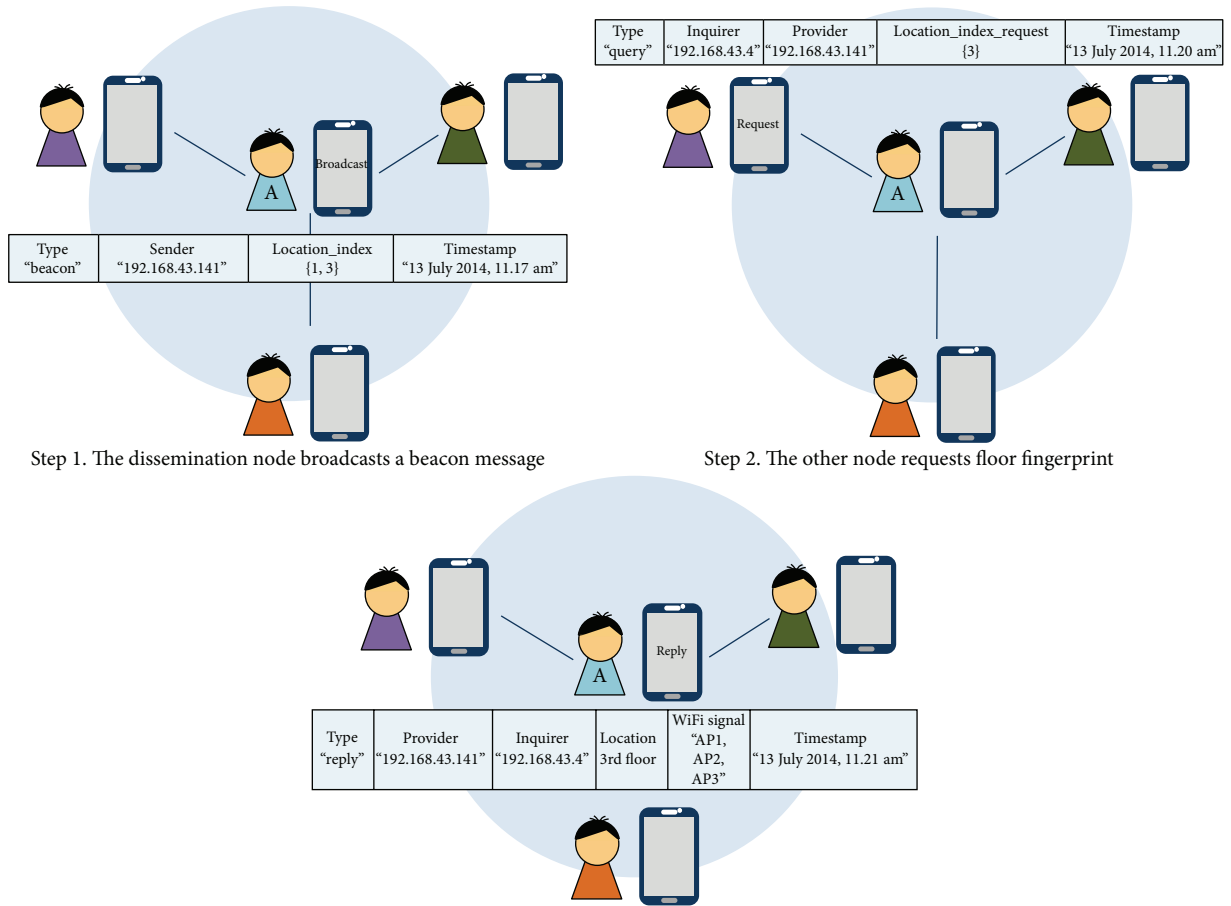


FIGURE 7: Fingerprint dissemination.

TABLE 1: Description of fields in the message.

Field name	Description
Type	A type of this message
Provider	IP address of dissemination node
Inquirer	IP address of node requesting floor fingerprints from dissemination node
Location_index	A list of floor fingerprints valid in fingerprint database of dissemination node
Location_index_request	A list of requested floor fingerprints
Location	A floor level when the dissemination node provides fingerprint to requesting node
WiFiSignal	A list of access points of requested floor fingerprint
Timestamp	Time that message was generated

query message to the dissemination node. The dissemination node sends a reply message to the node that requests floor fingerprints.

There are three types of messages disseminated in FloorLoc-SL: beacon message, query message, and reply message. The formats of beacon message, query message, and reply message are shown as in Steps 1–3 in Figure 7, respectively. All fields in message format will be described in Table 1.

4.5.2. Virtual Ad Hoc Network for Fingerprint Dissemination.

The virtual ad hoc network uses a Wi-Fi hotspot existing on the mobile phone to create an infrastructureless network the same as mobile ad hoc network. The mobile phone in the network has to enable a hotspot mode and wait for incoming connection. The mobile phones can connect to each other without any supported infrastructure. The Wi-Fi hotspot is chosen for the virtual ad hoc network because the Wi-Fi hotspot mode is enabled for all models of Android smartphones. Comparing with a pure version of mobile ad hoc network (MANET), MANET can be performed on some models of rooted smartphones (e.g., Google Nexus One with CyanogenMod). The Wi-Fi hotspot is more flexible. It can be implemented in the application layer without rooting the smartphones.

We have tested the Wi-Fi hotspot on the mobile phone as a preliminary study. We found that there are abilities and limitations of the Wi-Fi hotspot. A client can broadcast message to the hotspot and other clients. The hotspot can also broadcast message to all clients. Thus, the broadcasting message works completely in this network. However, there is an issue on unicasting. Clients can send message directly to each other. The hotspot can send direct message to a client, but the client cannot send direct message to the hotspot. In order to address the issue of unicasting, we applied broadcasting for sending unicast data. The mobile phone sends the message by broadcasting to all nodes. The nodes which received the message will check an IP address of receiver identified in the message. The message is discarded if the node is not the receiver. If the node is the receiver, it stores the message.

We designed the virtual ad hoc network for the fingerprint dissemination module. The working step is shown as flowchart in Figure 8. The mobile phone looks for the Wi-Fi hotspot named “LocalizationAP” when it enters into

fingerprint dissemination mode. In the case when the mobile phone detects *LocalizationAP*, it connects to the hotspot. The mobile phone begins fingerprint dissemination process by broadcasting a beacon message to all nodes in the network including the hotspot. The other nodes will request floor fingerprints using the fingerprint dissemination protocol. In the case when the mobile phone does not detect the *LocalizationAP*, the mobile phone switches to hotspot mode and waits for incoming connection.

5. Experiment and Discussion

In this section, four main issues are evaluated and discussed: floor localization, activity classification, floor counting, and overall system.

5.1. Floor Localization

5.1.1. Experiment Setting. The Wi-Fi signals were collected from real-world environment using Android OS version 2.2 based Google Nexus One. We developed an application on Android smartphone for collecting Wi-Fi scanning lists. The experimental site is 17th–20th floor of Charoen Visawakarma building in Faculty of Engineering, Chulalongkorn University. Figure 9 shows floor plans of 17th–20th floor and data collection traces. The red dots represent locations of existing Wi-Fi access points installed in the experimental site. Since Wi-Fi access points in the building provide unstable signals, the signal characteristic can change all the time. This situation affects a Wi-Fi scanning result. Therefore, we collected Wi-Fi scanning results in 12 days. There are 496 samples of Wi-Fi scanning lists for evaluation performance of floor localization algorithm.

5.1.2. Metrics. There are two metrics for considering the performance of our proposed floor localization algorithm.

(i) *Localization Accuracy.* This metric is used for evaluating the precision of our floor localization algorithm. It is a percentage of correct results compared with all results.

(ii) *Unknown Location Occurrence.* There are the cases that our floor localization algorithm provides unknown localization results. This metric measures the occurrences of “unknown.” It is percentage of unknown localization results

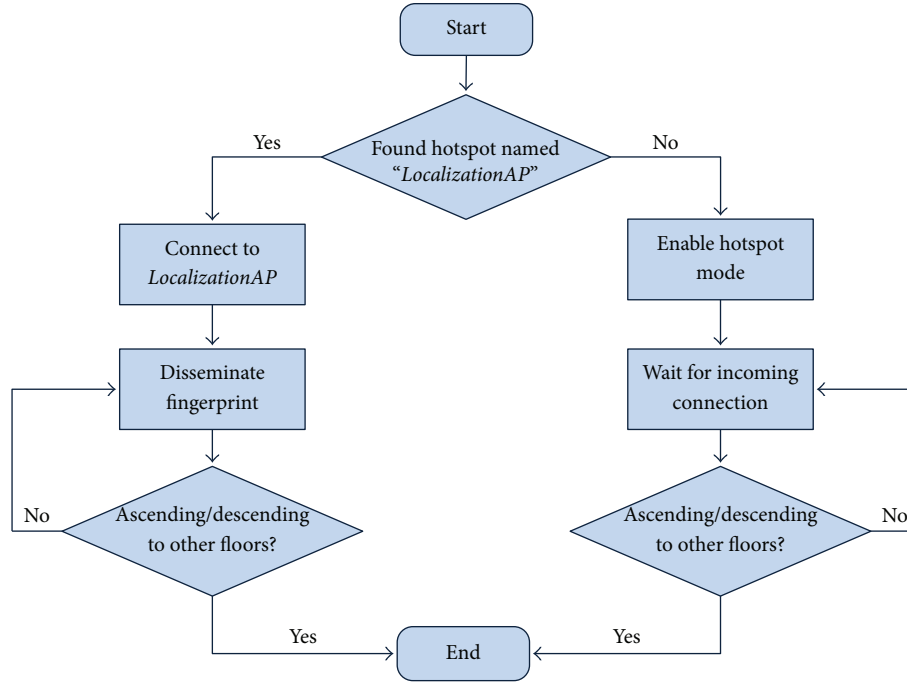


FIGURE 8: Flow chart of fingerprint dissemination via virtual ad hoc network.

compared with all results from the floor localization algorithm.

5.1.3. Results. The purpose of this experiment is to evaluate performance of our proposed floor localization algorithm. Fingerprints of all floors in the experimental site were known by the site survey. The number of top- N access points was varied from 1 to 5 to analyze the effect from this parameter.

The histogram in Figure 10 is localization accuracy of the floor localization algorithm. The x -axis is the number of top- N access points used as floor fingerprints. The y -axis is an accuracy of the algorithm. From the histogram, the maximum accuracy is 85%. The algorithm achieves the maximum accuracy when using top-3 access point as floor fingerprints.

The histogram in Figure 11 shows unknown location occurrence from the floor localization algorithm. The minimum unknown localization result is 3% when using top-1 access points as floor fingerprint. Using top-1 access point achieves 83% of accuracy. Using top-3 access points as floor fingerprint that provides the maximum localization accuracy of 85% provides 10% of unknown location occurrence.

From the experiment result, we found that using more top- N access points as a floor fingerprint reduces accuracy of our proposed floor localization algorithm. This is because the Wi-Fi signal from access points installed in an adjacent floor can interfere to the current floor. When the mobile phone scans Wi-Fi signal, it includes access points transmitting the signal interferences in the top- N access points. With these top- N access points, the floor fingerprint is incorrect and

affects localization result. The example of this situation can be illustrated in Figure 12 for more clarification.

Figure 12(a) shows an example building installed Wi-Fi access points. The Wi-Fi scanning results of each floor are displayed in Figure 12(b). From Figure 12(b), you will see that Wi-Fi signal from the access points in Figure 12(a) can interfere to adjacent floor. The mobile phone can hear their signals even if the signal strength is low. If the fingerprint algorithm creates a floor fingerprint with top-5 access points, the floor fingerprints of 1st floor and 2nd floor are the same as shown in Figure 12(c). With these floor fingerprints, the algorithm cannot determine the current floor of mobile phone. If the floor fingerprint is created from top-3 access points, the fingerprints of 1st floor and 2nd floor are different as shown in Figure 12(d). The floor localization algorithm can provide a current floor.

5.2. Activity Classification

5.2.1. Experiment Setting. We collected data from accelerometer sensor and barometer sensor to evaluate performance of the activity classification by the acceleration approach and by the atmospheric pressure approach. In this experiment, we developed an Android application to record the data from accelerometer and barometer sensor. The data were recorded from 5 users using Samsung Galaxy S4 while they were performing activities: steady, walking, and going up/down the stairs with controlling the phone position and orientation. The users carried their mobile phone by their hand while data were being recorded and labeled their current activity states. The data sampling frequency was 7.0 Hz.

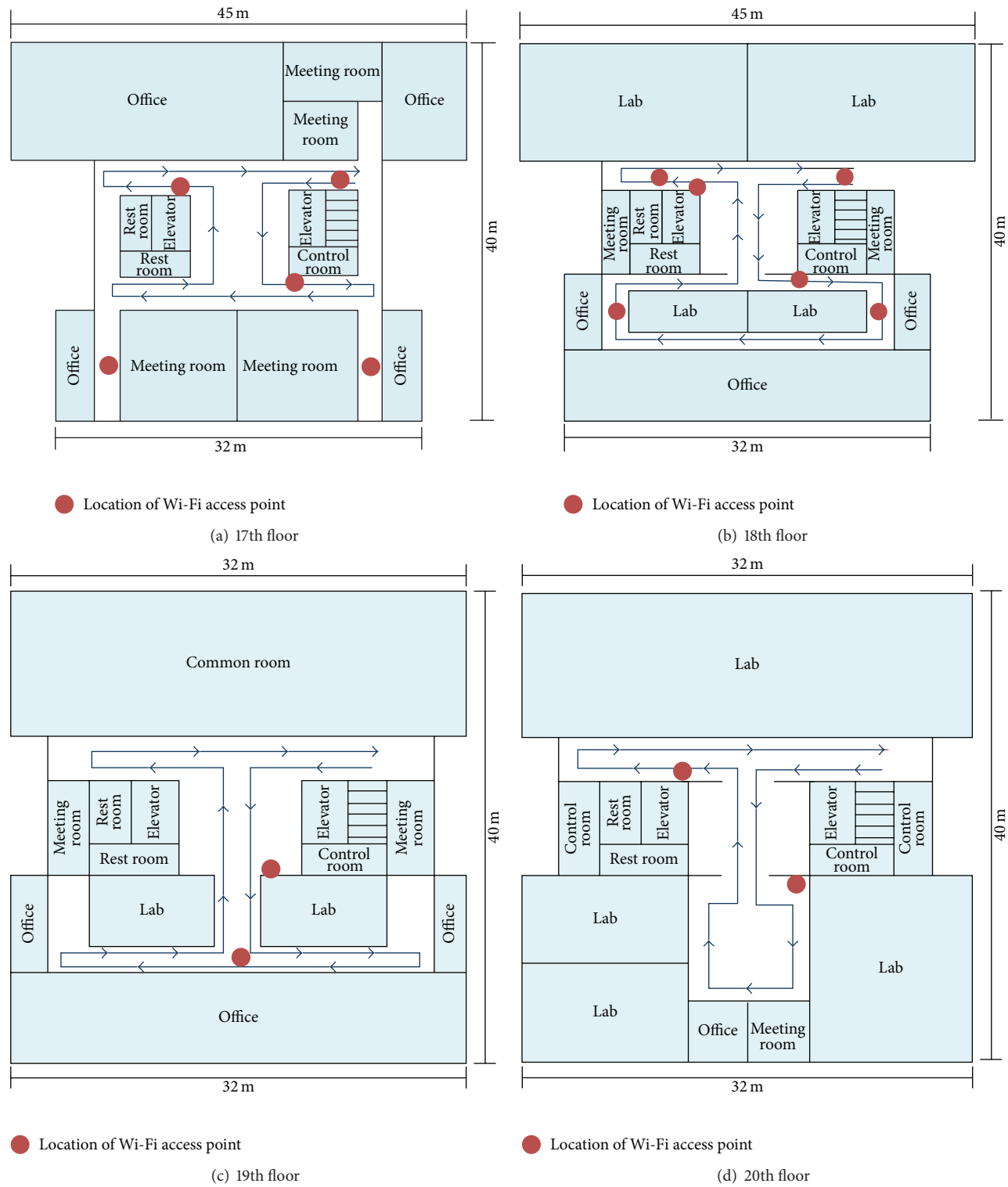


FIGURE 9: Floor plans of the experimental site and data collection traces.

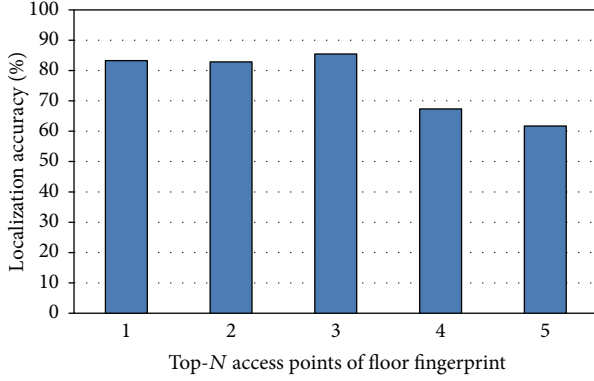


FIGURE 10: Accuracy of our proposed floor localization algorithm.

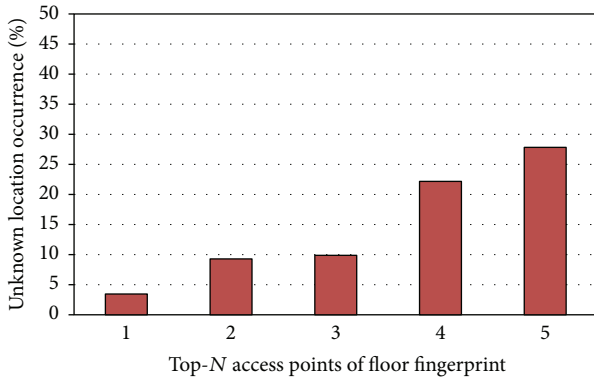


FIGURE 11: Unknown location occurrence of our proposed floor localization algorithm.

5.2.2. Metrics

Accuracy of Activity Classification. This metric is a percentage of correct activity classification results compared with all results. The metric represents the performance of our proposed activity classifications.

5.2.3. Results. In this experiment, we evaluate activity classification with both acceleration approach and atmospheric pressure approach, study the effect from window size in the algorithm, and compare the performance of these approaches.

From the experiment, we found that each user has personal thresholds (α_{walk} , α_{up} , and α_{down}) for classification of their activities by the acceleration approach. This is because movements are different in each person. Table 2 shows thresholds of each user in the experiment. The accuracy of the activity classification with the acceleration approach is shown as a histogram in Figure 13. The results in the histogram are average accuracy from result of all users in the experiment and distinguished by a kind of activity. The x -axis is the window size that is used in the activity classification algorithm.

From the histogram in Figure 13, we found that using larger window size in the activity classification algorithm provides better accuracy of steady and walking detection

TABLE 2: Thresholds in activity classification algorithm using acceleration approach of each user.

	Thresholds		
	α_{walk}	α_{up}	α_{down}
User 1	0.4	2.0	3.0
User 2	0.6	2.2	4.1
User 3	0.9	3.8	5.1
User 4	0.6	3.8	5.0
User 5	0.8	3.3	6.0

but reduces accuracy of up and down detection. This is because the algorithm requires certain amount of time to perform acceleration data sampling and fills the result in the window for majority voting. The duration of going up/down stairs from a floor to the stair landing is short. The previous activity states (walking, steady) and activity state of going up/down the stairs are possibly filled in the same window. With this situation, the activity states “up” and “down” are not a majority of the window and affect activity classification result. Although the majority voting is applied to reduce noise of activity classification using acceleration approach, sometimes, it provides a noise or an incorrect activity state as a result when noise is majority of the window.

Figure 14 is accuracy of the activity classification with the atmospheric pressure approach distinguished by activity. In this experiment, we tested all possible values of thresholds for the activity classification algorithm using this approach and we found that setting -0.04 for β_{up} and 0.04 for β_{down} provides the best accuracy. These thresholds are efficient to activity classification for all users moving in the prototype building. The histogram in Figure 14 shows the accuracy of the activity classification algorithm using this approach. The x -axis represents window size used in the algorithm. The results in the histogram are average result of all five users in the experiment.

From the accuracy in Figure 14, using larger window size in the activity classification algorithm (Algorithm 4) provides more accuracy, especially detecting going up the stairs and going down the stairs. Considering the graph of atmospheric pressure in Figure 5, there are fluctuations in the data sampling from barometer sensor. The algorithm (Algorithm 4), which averages all atmospheric pressure p_i in the window, reduces the fluctuation. Therefore, the accuracy of activity classification is improved when using larger window size. However, when the window size goes too large, the accuracy decreases. As shown in Figure 14, when the window size is 30, the accuracy starts to drop. This is because when the window size is too large, many activities happen within one window and affects the average pressure of windows in the activity classification using atmospheric pressure approach. This situation causes the algorithm not to be able to provide the result efficiently.

Figure 15 is the accuracy comparison of the activity classification using the acceleration approach and that using the atmospheric pressure approach. These results are average accuracy of all activities classification. The maximum

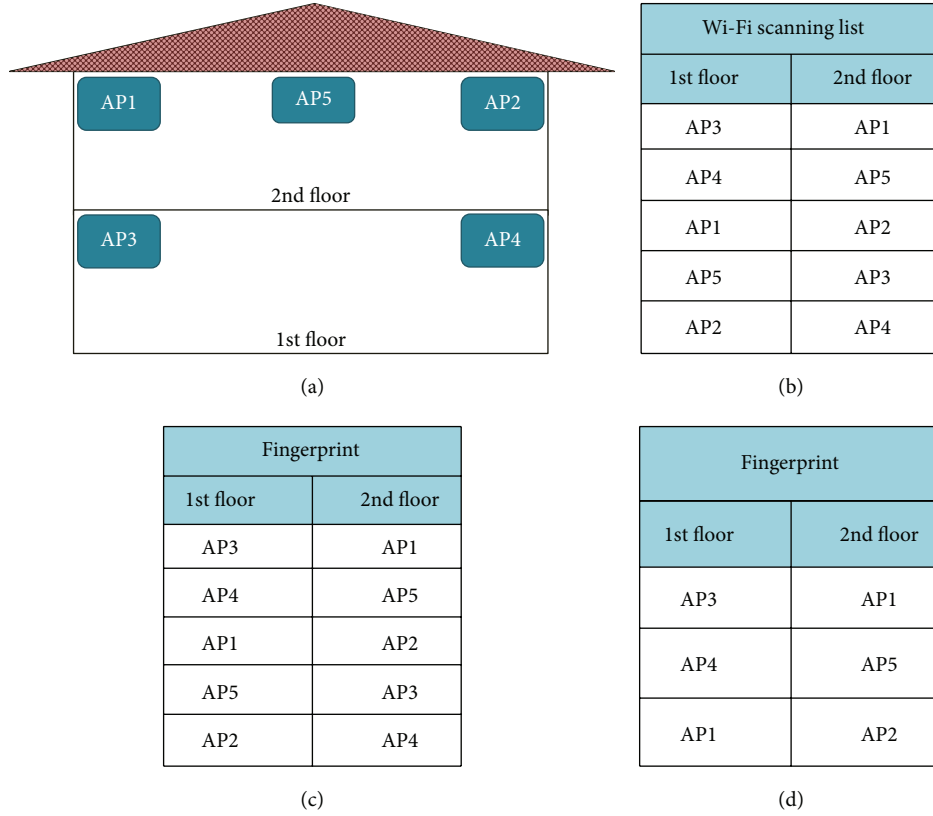


FIGURE 12: The example scenario explains why using more top- N access point for creating a floor fingerprint reduces localization accuracy.

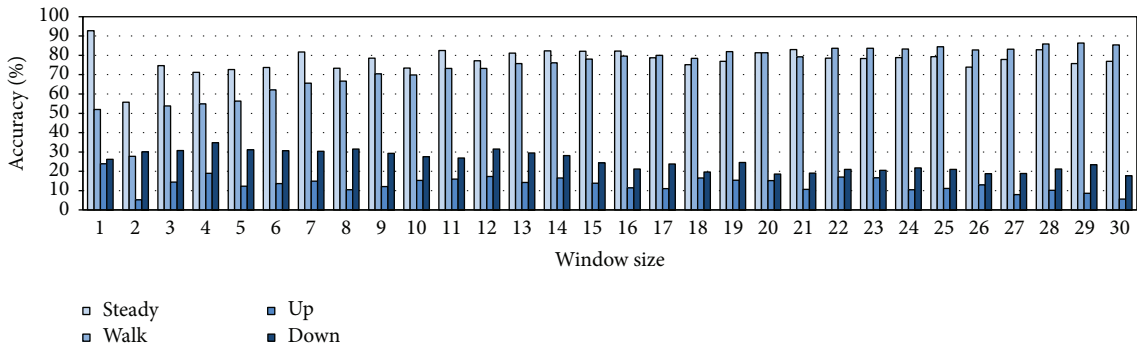


FIGURE 13: Activity classification accuracy of acceleration approach.

accuracy of the activity classification using the acceleration approach is 51% using the window size of 14 while the maximum accuracy of that using the atmospheric pressure approach is 85% using the window size of 29.

5.3. Floor Counting Module

5.3.1. Experiment Setting. The data were collected from accelerometer and barometer sensor using Samsung Galaxy S4 with the Android application we mentioned in Section 5.2. There were 5 users for collecting the data in the prototype building. The users walked to other floors using the stairs in the building and labeled their activity history. The position

and orientation of the mobile phone was controlled like the previous experiment. The data were sampled in the rate of 70 Hz.

There are 5 accelerometer traces of going up and 5 accelerometer traces of going down the stairs one floor to evaluate performance of the floor counting by the acceleration approach. We evaluated only one-floor scenario because we need to ensure that the state machine in Figure 6 can detect when the user goes up/down the stairs and update the parameter *FLOOR_COUNT*. We have 20 barometer traces to discover optimal thresholds and evaluate performance of the floor counting by the atmospheric pressure approach. These traces are going up/down the stairs one floor and two floors.

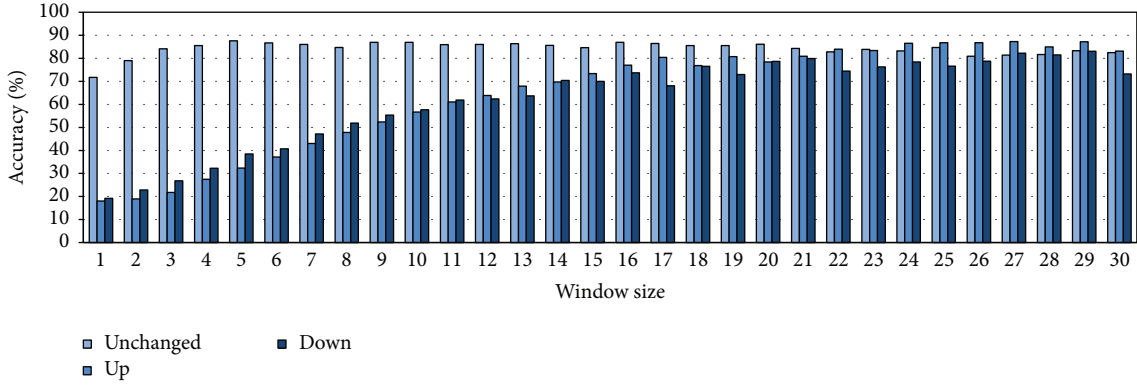


FIGURE 14: Activity classification accuracy of atmospheric pressure approach.

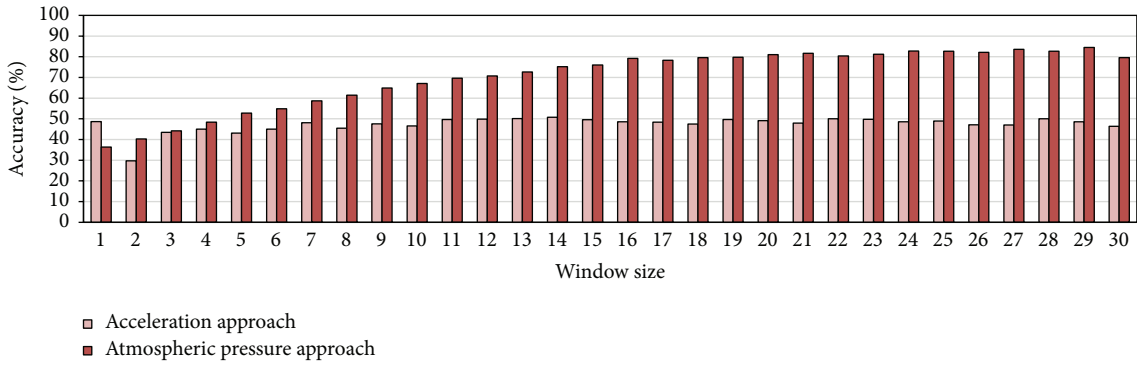


FIGURE 15: Comparison of average accuracy of activity classification.

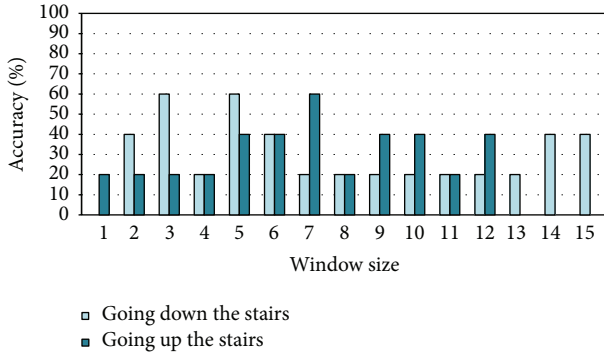


FIGURE 16: Accuracy of floor changing detection.

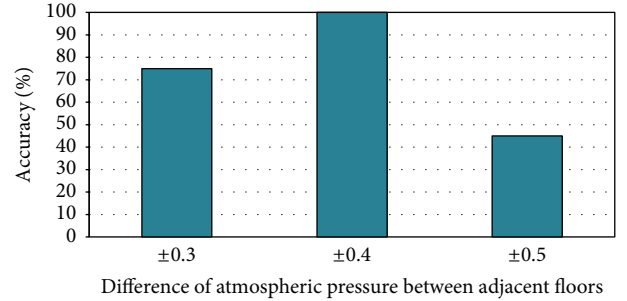


FIGURE 17: Threshold evaluation result for floor counting methodology by atmospheric pressure.

5.3.2. Results. The histogram in Figure 16 is the accuracy of detection when the user goes up/down the stairs using the state machine for the floor counting with the acceleration approach. The x -axis is a window size in the activity classification algorithm by the acceleration approach. Considering using the window size of 14 that provides the best accuracy of the activity classification (51%), it can detect going down the stairs with 40% of accuracy but it cannot detect going upstairs at all. Using the window size of 5 can provide the best detection performance for going up/down the stairs (40% and 60% resp.) but it can provide 43% of accuracy for the activity classification as shown as in Figures 13 and 15.

We have to find the thresholds: $\beta_{\text{floor_up}}$ and $\beta_{\text{floor_down}}$ for the floor counting algorithm by the atmospheric approach before evaluating performance. In the experiment, we varied the different atmospheric pressure between adjacent floors

from 0.1–1.0 hPa. The results are shown in Figure 17. There are only three difference values (± 0.3 , ± 0.4 , and ± 0.5) providing correct results. From the histogram, we found 100% of the floor counting accuracy is achieved if the floor counting algorithm uses -0.4 and 0.4 for $\beta_{\text{floor_up}}$ and $\beta_{\text{floor_down}}$, respectively.

We collected three barometer traces of going up and three barometer traces of going down the stairs three floors and tested the floor counting algorithm by these traces. The algorithm still achieves 100% of accuracy. Moreover, the floor counting by the atmospheric pressure approach can work with other scenarios such as the user goes to other floors by an escalator or an elevator. This is because the different atmospheric pressure between any adjacent floors is constant (0.4 for our prototype building) and depends on the height between floors. The atmospheric pressure of each floor in

TABLE 3: Summary differences between acceleration approach and atmospheric pressure approach.

	Approach	
	Acceleration	Atmospheric pressure
Sensor	Accelerometer	Barometer sensor
Embedded on	All smartphones	Flagship smartphones
Parameters	α_{walk} , α_{up} , and α_{down}	β_{up} and β_{down}
Parameter calibration	Individual user	Each building
Horizontal activity discrimination	"steady" and "walk"	"unchanged"
Vertical activity discrimination	"up" and "down"	"up" and "down"
Activity classification accuracy	51% (window size = 14)	85% (window size = 29)
Floor counting scenario	One stair-landing scenario	Stairs, escalator, and elevator
Floor counting accuracy	50% (window size = 5)	100% ($\beta_{\text{floor_up}} = -0.4$, $\beta_{\text{floor_down}} = 0.4$)

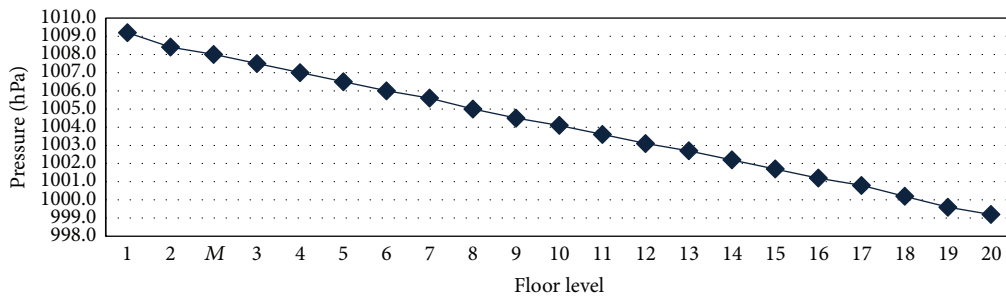


FIGURE 18: Atmospheric pressure of each floor in the experimental building.

our experimental building is shown in Figure 18. Velocity of mobile phone user movement does not affect our proposed algorithm. Our application on the Android phone can collect data from barometer sensor with 7.0 Hz of sampling rate. In general, the height of consecutive floor is about 3 meters. It means that our algorithm can work properly as long as the speed of mobile phone user is not higher than 21 m/s.

As a result, the differences of the activity classification and the floor counting using the acceleration approach and those using atmospheric pressure approach are concluded in Table 3. From Table 3, although the activity classification and the floor counting using the atmospheric pressure approach overcome those using the acceleration approach, the atmospheric pressure approach has some limitation. The barometer sensors are often embedded on flagship model of smartphone while the accelerometers are embedded on almost models of smartphone. However, our proposed floor localization system has fingerprint dissemination mechanism. If there is at least one mobile phone in the virtual network embedded with the barometer sensor, it will share the reliability of the floor fingerprints to others. The reliability of fingerprint will be considered in the future work.

5.4. Overall System

5.4.1. Experiment Setting. The objective of this experiment is to evaluate the performance of our localization algorithm

with the fingerprint self-learning and the fingerprint dissemination algorithm. In this experiment, we implemented a simulation using Java to evaluate the performance of our proposed system. The simulation focuses on the fingerprint self-learning and fingerprint dissemination. We assume the activity classification results are totally correct. We developed the Android application to collect data for this experiment. The data are a Wi-Fi scanning result, a current floor level, a current activity state, and an activity history of mobile phone user. Seven user traces with their actual activities were collected in the prototype building. Each trace represents a node, so there are seven nodes in the simulation. We simulated the experiment for 3 times.

5.4.2. Results. The histogram in Figure 19 shows the localization accuracy of our proposed system. The x -axis is the number of exchanging nodes that are sharing the fingerprints to each other. Without the fingerprint dissemination, our proposed system achieves 33% of localization accuracy. From the histogram, we found that exchanging the fingerprints with more number of nodes can improve the localization accuracy. The maximum accuracy of 87% is achieved when receiving the floor fingerprints from four exchanging nodes and using top-5 access points in the process of the floor localization.

According to Figure 19, at a particular number of access points used in the process of the floor localization, there is

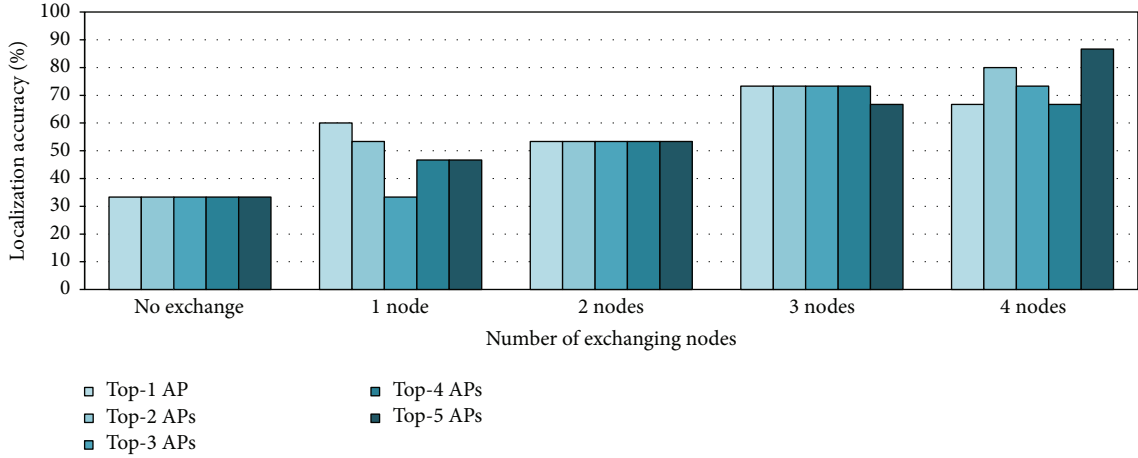


FIGURE 19: Accuracy of the localization system with fingerprint self-learning and fingerprint dissemination.

a tendency that the more the number of exchanging nodes, the more the accuracy of localization. However, there are some cases when receiving the floor fingerprints from more exchanging nodes cannot guarantee that the system achieves more accuracy of localization. Considering using top-1 access points, receiving the fingerprints from one node provides 60% of localization accuracy while receiving the fingerprints from two exchanging nodes achieves only 53% of localization accuracy. This is because, sometimes, the fingerprint self-learning module provides a fingerprint with wrong location information. This node can propagate the wrong fingerprint to others. As a consequence, the localization accuracy is reduced.

Considering the case when the mobile phone exchanges floor fingerprints with 4 nodes in Figure 19, using top-5 access points as a floor fingerprint provides best accuracy. This is because, in the simulation, we found that many localization processes and fingerprint self-learning mechanism are performed at the position near the stairs. The stairs are the location that is most likely to be interfered with Wi-Fi signals from access points installed in adjacent floors. At the stairs, we found many cases in which access points installed near the stairs area of adjacent floors provide signals stronger than access points of the current floor. Using less top- N access point as a floor fingerprint has a possibility that the floor fingerprint is same as fingerprint of adjacent floors. This causes floor localization algorithm to be not able to determine the current floor of mobile phone. Using more top- N access points as a floor fingerprint can make sure that the fingerprint includes access points of this floor even though their signals are weaker than signal of access points in adjacent floors. The floor fingerprints of adjacent floors will be different.

5.4.3. Comparison with Other Works. Many floor localization systems have been proposed. Table 4 compares our proposed FloorLoc-SL with the previously proposed floor localization systems and a system proposed by Marques et al. [12], SkyLoc [13], and FTrack [14].

In Table 4, all systems are floor localization systems developed for mobile phone platform. The system proposed by Marque et al. provides 99.5% of floor localization accuracy, which is the maximum accuracy. The SkyLoc achieves up to 73% of floor localization accuracy. Although the complexity of these floor localization algorithms is medium level, in practical, the fingerprint map has a lot of sample points to be processed. Moreover, both of them require site survey that leads to labor cost and time consumption. FTrack is a floor localization algorithm that does not require any site survey. The floor localization accuracy of FTrack is 90%, but its algorithms are high complexity. The system needs back-end server to support processing. Our proposed FloorLoc-SL achieves comparable floor localization accuracy (87%) with low cost because it does not require any site survey and back-end server. Moreover, our proposed floor localization algorithm is the simplest algorithm that appropriates to be processed on any mobile devices. Therefore, our proposed system can flexibly operate in any buildings and any mobile devices.

6. Conclusion

In this paper, we proposed FloorLoc-SL: a floor localization system without site survey and supporting from any back-end servers. FloorLoc-SL consists of three main processes: floor localization, fingerprint self-learning, and fingerprint dissemination. The floor localization provides the current floor of the mobile phone using top- N access point from Wi-Fi scanning list. It achieves up to 85% of accuracy. The fingerprint self-learning uses the activity classification module and the floor counting module to detect the current floor in the case when the floor localization algorithm cannot provide the current location. We proposed two approaches for both the activity classification and the floor counting: acceleration approach and atmospheric pressure approach. The performances of the atmospheric pressure approach overcome those of the acceleration approach. In our system, moreover, the fingerprint can be shared by other mobile

TABLE 4: Comparison of the proposed floor localization systems.

Issue	Marques et al. [12]	SkyLoc [13]	FTrack [14]	FloorLoc-SL
Floor localization	✓	✓	✓	✓
Mobile phone platform	✓	✓	✓	✓
Fingerprint-based approach	✓	✓	✗	✓
Floor localization approach	Wi-Fi signal fingerprint	GSM signal fingerprint	Travelling times and step counting between any two floors	Wi-Fi signal fingerprint
Low cost due to no requirement of site survey	✗	✗	✓	✓
Algorithm complexity	Medium $O(nm)$ n = the number of floor levels m = the number of sample points	Medium $O(nm)$ n = the number of floor levels m = the number of sample points	High Grouping: $O(n^2 \ln n)$ Merging: $O(n^3 \ln n)$ n = the number of trail segments	Low Fingerprint: $O(N)$ Floor localization: $O(F)$ N = the number of top- N access points F = the number of floor levels
Accuracy	99.5%	73%	90%	87%

phones via virtual ad hoc network. This can improve the system accuracy from 33% to 87%.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by Ratchadapiseksomphot Endowment under Outstanding Research Performance Program and Special Task Force for Activating Research (STAR) Funding in Wireless Network and Future Internet Research Group, Chulalongkorn University. The scholarship from the Graduated School Chulalongkorn University to commemorate the 72nd anniversary of his Majesty King Bhumibol Adulyadej is gratefully acknowledged.

References

- [1] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, IEEE, Tel Aviv, Israel, March 2000.
- [2] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, pp. 385–386, ACM, Bethesda, Md, USA, July 2011.
- [3] Y. Zheng, W. Chenshu, and L. Yunhao, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 269–280, Istanbul, Turkey, August 2012.
- [4] Y. Sungwon, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: calibration-free crowdsourced indoor localization," in *Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 2481–2489, Turin, Italy, April 2013.
- [5] T. Higuchi, S. Fujii, H. Yamaguchi, and T. Higashino, "Mobile node localization focusing on stop-and-go behavior of indoor pedestrians," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1564–1578, 2014.
- [6] H. Buyruk, A. K. Keskin, S. Sendil et al., "RF fingerprinting based GSM indoor localization," in *Proceedings of the 21st Signal Processing and Communications Applications Conference (SIU '13)*, pp. 1–4, IEEE, Haspolat, Northern Cyprus, April 2013.
- [7] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: fine-grained indoor localization," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2210–2218, Orlando, Fla, USA, March 2012.
- [8] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [9] K. Khaoampai, K. Suksen, K. N. Nakorn, and K. Rojviboonchai, "DecaFriend: serverless indoor localization system on mobile phone platform," in *Proceedings of the 11th International Joint Conference on Computer Science and Software Engineering (IJCSSE '14)*, pp. 176–181, IEEE, Chonburi, Thailand, May 2014.
- [10] W. Chenshu, Y. Zheng, L. Yunhao, and X. Wei, "WILL: wireless indoor localization without site survey," in *Proceedings of the IEEE Conference on Computer Communications*, pp. 64–72, Orlando, Fla, USA, March 2012.
- [11] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proceedings of the 9th International Conference on Mobile*

- Systems, Applications, and Services (MobiSys '11)*, pp. 141–154, Bethesda, Md, USA, July 2011.
- [12] N. Marques, F. Meneses, and A. Moreira, “Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN '12)*, pp. 1–9, IEEE, Sydney, Australia, November 2012.
 - [13] A. Varshavsky, A. Lamarca, J. Hightower, and E. de Lara, “The skyLoc floor localization system,” in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, pp. 125–134, White Plains, NY, USA, March 2007.
 - [14] H. Ye, T. Gu, X. Zhu et al., “FTrack: infrastructure-free floor localization via mobile phone sensing,” in *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications (PerCom '12)*, pp. 2–10, IEEE, Lugano, Switzerland, March 2012.
 - [15] Y. Liu, M. Dashti, and J. Zhang, “Indoor localization on mobile phone platforms using embedded inertial sensors,” in *Proceedings of the 10th Workshop on Positioning, Navigation and Communication (WPNC '13)*, pp. 1–5, IEEE, Dresden, Germany, March 2013.
 - [16] K. Khaoampai, K. N. Nakorn, and K. Rojviboonchai, “Low complexity floor localization algorithm for mobile phone,” in *Proceedings of the 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '14)*, pp. 1–6, Nakhon Ratchasima, Thailand, May 2014.
 - [17] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
 - [18] T. Brezmes, J.-L. Gorricho, and J. Cotrina, “Activity recognition from accelerometer data on a mobile phone,” in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, S. Omatu, M. Rocha, J. Bravo et al., Eds., vol. 5518 of *Lecture Notes in Computer Science*, pp. 796–799, Springer, Berlin, Germany, 2009.
 - [19] J. Yang, “Toward physical activity diary: motion recognition using simple acceleration features with mobile phones,” in *Proceedings of the 1st ACM International Workshop on Interactive Multimedia for Consumer Electronics (IMCE '09)*, pp. 1–9, ACM, Beijing, China, October 2009.
 - [20] A. M. Khan, Y. K. Lee, S. Y. Lee, and T. S. Kim, “Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis,” in *Proceedings of the 5th International Conference on Future Information Technology (FutureTech '10)*, pp. 1–6, IEEE, Busan, South Korea, May 2010.

