

RESEARCH

Open Access



# Wi-Fi indoor positioning and navigation: a cloudlet-based cloud computing approach

Tran Trong Khanh, VanDung Nguyen, Xuan-Quy Pham and Eui-Nam Huh\*

\*Correspondence:  
johnhuh@khu.ac.kr  
Department of Computer  
Science and Engineering,  
Kyung Hee University,  
1732 Deokyoungdaero,  
Giheung-gu, Yongin-si,  
Gyeonggi-do 17104, South  
Korea

## Abstract

Wi-Fi-based indoor positioning for determining accurate wireless indoor location information has become crucial in meeting increasing demands for location-based services by leveraging the Internet of Things (IoT) and ubiquitous connectivity. Most Wi-Fi-based indoor positioning techniques using wireless received signal strength (RSS)-based methods are affected by the indoor environment and depend on the respective signals from at least three reference access points. In this paper, we propose a cloudlet-based cloud computing system enabling Wi-Fi indoor positioning and navigation through a Wi-Fi located on a one-hop wireless network. Our cloudlet-based cloud computing system provides the reference point data and real-time interactive response for a self-driving indoor cart. The system was tested in a real environment with the following results: (1) our system autonomously performed actions, such as turning right or left or going straight according to a movement decision algorithm and determined the position within a stable range of Wi-Fi coverage; (2) the cloudlet and core cloud can track navigation for an indoor self-driving cart; (3) the global and local positions designed for reference access points and a specific position can navigate the self-driving cart to a particular position accurately; (4) the moving edge clouds play a role in deciding three action movements (go straight, turn left, and turn right), as well as managing the local position of the items; and (5) a core cloud is deployed to store all information for the items, such as their positions and corresponding Wi-Fi locations. A core cloud manages items that have the same position (i.e., a global position) defined as the corresponding Wi-Fi location. Finally, the practical results have significance in designing a cloudlet-based cloud computing system enabling Wi-Fi indoor positioning and navigation.

**Keywords:** Indoor localization, Location-based services, Cloudlet, RSSI, Wi-Fi, Internet of Things

## Introduction

Increasing demands for location-based services require accurate wireless indoor location information. Location-based services include indoor navigation for people or robots, personnel, asset tracking, guiding blind people, factory automation, workplace safety, locating patients in a hospital, and location-based advertising [1]. Additionally, such services are becoming essential in various other fields such as mobile commerce, parcel or vehicle tracking, discovering the nearest shops or restaurants, and social

networking [2]. Moreover, analyses of personnel and asset tracking and collecting movement data have been limited.

With the increasing prevalence of global positioning system (GPS) applications, satellite signals have had a significant impact on outdoor positioning systems owing to their high accuracy. However, in indoor environments, indoor location and navigation remain unsolved problems. Because of such factors as multipath effects and Wi-Fi signal interference, satellite signals become unreliable for positioning indoors. Consequently, GPS-based indoor positioning techniques still face major obstacles, including the unavailability or degradation of GPS signals, real-world indoor environments, and low-grade devices [3, 4]. Given such circumstances, various positioning techniques have been proposed. According to [5], positioning techniques are classified into two categories, infrastructure-based and infrastructure-free technologies. An infrastructure-based technology requires pre-installation and configuration of specialized hardware in the environment. For instance, these include radio-frequency identification (RFID) [6], ultrasound [7], Bluetooth [8], ultra-wideband beacons (UWB) [9], ZigBee [10], infrared [11], and pseudolites [12]. Infrastructure-free technologies are typically based on Wi-Fi [13–16], magnetic fields, motion sensors [(inertial measurement units (IMUs)], and vision techniques [17]. Recently, there are many other technologies like audio signal based localization [18], magnetic field-based localization [19]. Compare to RF methods, audio signal based localization or acoustic localization is more accurate and cheaper [20]. However, acoustic localization requires microphones and speakers which are available in every smart mobile device to combine with Bluetooth low energy (BLE) and Wi-Fi-based approaches. In this system, BLE or Wi-Fi can be utilized for rough location estimation and acoustic signals are used for computing the precise location [20].

Wi-Fi based indoor localization systems have become a prominent tool for indoor positioning for various reasons. First, nearly all smartphones have a built-in Wi-Fi module. Second, Wi-Fi access points are installed ubiquitously, and a Wi-Fi based indoor localization system has suitable cost and accessibility [2, 21]. Third, Wi-Fi does not require additional special-purpose hardware. Location estimation can be easily estimated by measuring the received signal strength (RSS) values from a Wi-Fi access point. Finally, the bandwidth of Wi-Fi systems has increased significantly to meet the requirements of high data rates [1].

Alternatively, Wi-Fi indoor positioning techniques are classified into two categories, signal propagation models [17, 32] and location fingerprinting [29]. A comparison of signal propagation models and the location fingerprinting method is presented in Table 1. In a signal propagation model, an indoor positioning system using the time-of arrival (ToA) and time difference of arrival (TDoA) suffers from multipath fading on several paths [5] while measuring the distance to the station from mobile devices. Alternatively, the location can be estimated using the angles of the signals received from the mobile user and the Wi-Fi access point; this includes the angle of arrival (AoA) and angle of departure (AoD) techniques. Most systems based on AoA measure the relative angles between signals coming from multiple anchor nodes to estimate the position requiring the antenna directions of both the mobile user and Wi-Fi access point to be known. By measuring the time-of-flight of the signal traveling from the sender to the user and back, the round-trip time-of-flight (RToF) of the signal is used to measure a location.

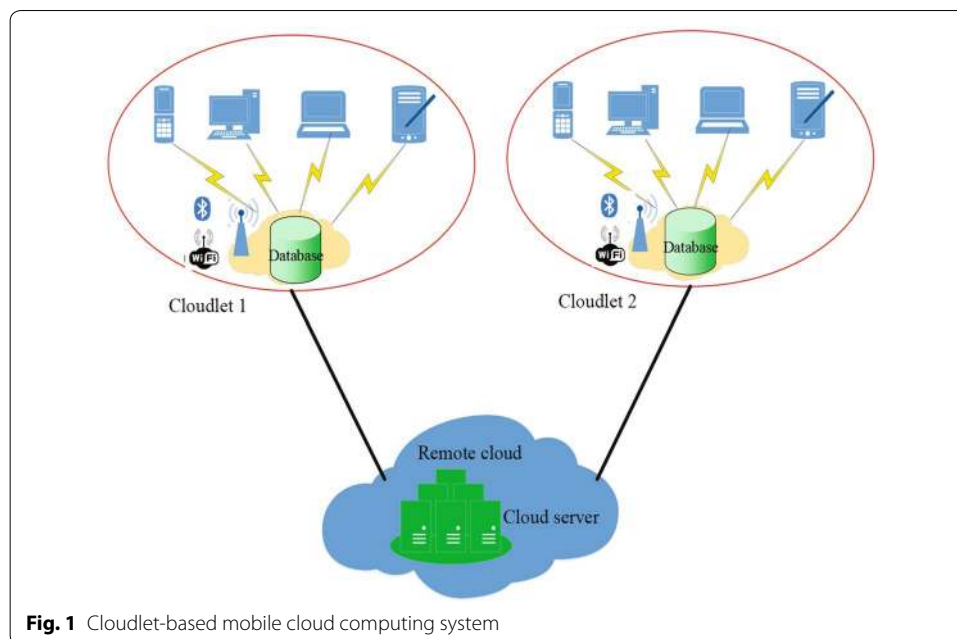
**Table 1 Comparisons of signal propagation model and location fingerprinting method**

	Signal propagation mode	Location fingerprinting method
Technique	RSS [2, 21], ToA [22–24], TDoA [25], AoA [26, 27], AoD [28]	Fingerprinting [29]
Measurements	Signal strength, angles	Fingerprint database
Accuracy	Medium	High
Time cost	Low	High
Distance	Low	Medium
Algorithm	Deterministic	Deterministic and probabilistic
Specification	Time based	Range based
Advantages	Additional special-purpose hardware is not required Acceptable cost and accessibility	Addresses some of the problems related to non-line-of-sight and multipath propagation [29] Does not require exact locations of access points [30]
Disadvantages	More difficult regarding synchronization of receiver/transmitter-side times and accuracy of short-range flight time measurements For angle calculation, complex hardware is required to determine the angle of arrival Poor positioning accuracy in real-world indoor environments because of a complex accurate propagation model [30, 31] Multipath is an issue with signal propagation	Time-consuming and laborious Accurate fingerprint data depends on the autonomic construction of a fingerprint database When real-world indoor environments change significantly, the fingerprint database has to be rebuilt [29]

However, this requires the exact delay and processing time. In location fingerprinting, a database containing measurements of the wireless signals at various reference points in a wireless LAN coverage area is established first. Indoor positioning systems using location fingerprinting compare the wireless signal measurements with the reference data [29]. However, this method requires database generation and maintenance. Compared with location fingerprinting, implementing signal propagation is simple. However, signal propagation (as a result of such factors as penetration losses through walls and floors) and multipath propagation are still very complicated [29]. For this reason, a novel Wi-Fi-based indoor positioning system was proposed to achieve a better performance.

In addition, in the era of computing paradigms, cloudlet is known as the technology at the edge of the Internet for deploying mobile cloud services. The aim of using cloudlets, typically accessed through Wi-Fi connections, is to bring cloud technologies closer to the end-user and provide resource- and latency-sensitive applications [33]. Moreover, cloudlets are small-scale data center that are designed to provide cloud computing applications quickly to mobile devices such as smartphones, tablets, and wearable devices within close geographical proximity. This places cloudlets in the following three-tier hierarchy: mobile device, cloudlet, and remote cloud, as shown in Fig. 1. Based on the mobile user's request, one or more custom virtual machines can be instantiated immediately on the cloudlet for remote execution of applications [34]. The advantages of cloudlets include the following:

1. Through Wi-Fi located on a one-hop wireless network [35], a cloudlet system efficiently provides a powerful computing resource and speeds up mobile application executions.
2. The real-time interactive response can access the cloudlet through a one-hop high-bandwidth wireless to reduce the transmission delay [35, 36].

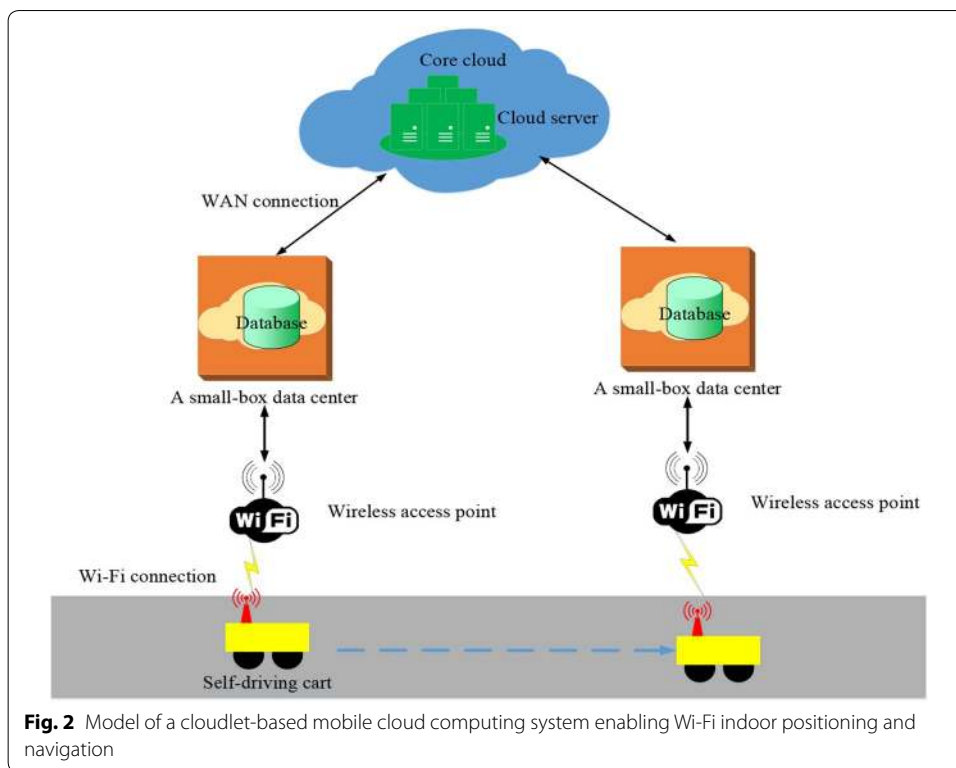


**Fig. 1** Cloudlet-based mobile cloud computing system

3. Utilizing mobile device connectivity to nearby cloud servers enables a cloudlet to overcome the distant wide-area network latency and cellular energy consumption problems [37].
4. Using cloudlets is more optimized and efficient, enhancing the user experience when computation-intensive tasks offload to nearby cloud servers in a cloudlet-based cloud computing system [38].
5. Cloudlets leverage the computational capacity of connected mobile devices [39].

Cloudlets deployed one wireless hop away from mobile devices can process the computationally intensive tasks offloaded from devices efficiently [33]. Therefore, cloudlets are typically set up at a public place, such as a shopping center, theater, office building, or assembly room, to enable convenient access for mobile devices [35]. Compared to the baseline Wi-Fi indoor positioning system, a combination of a cloudlet-based cloud computing system, indoor positioning, and navigation, considered as a single system, is practical. To achieve that, we designed a model of a cloudlet-based mobile cloud computing system enabling Wi-Fi indoor positioning and navigation, as shown in Fig. 2. The system consists of a self-driving cart, a small-box data center (cloudlet) available in a wireless access point, and a core cloud.

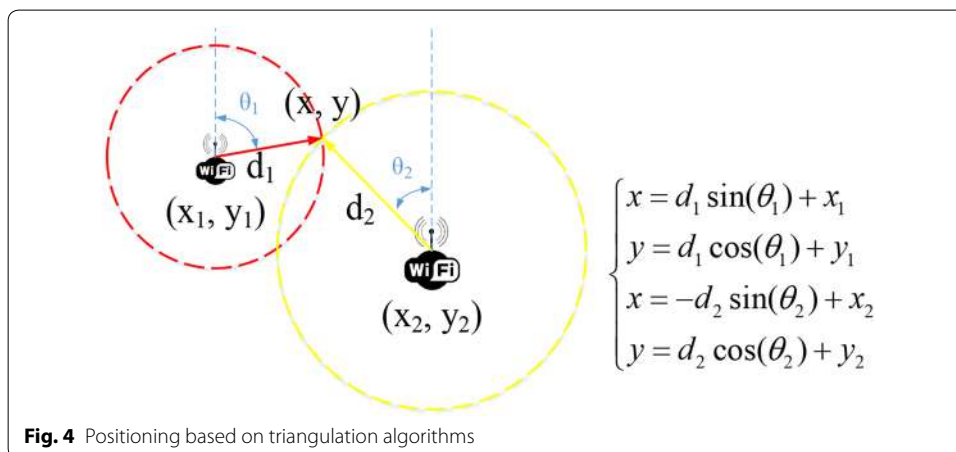
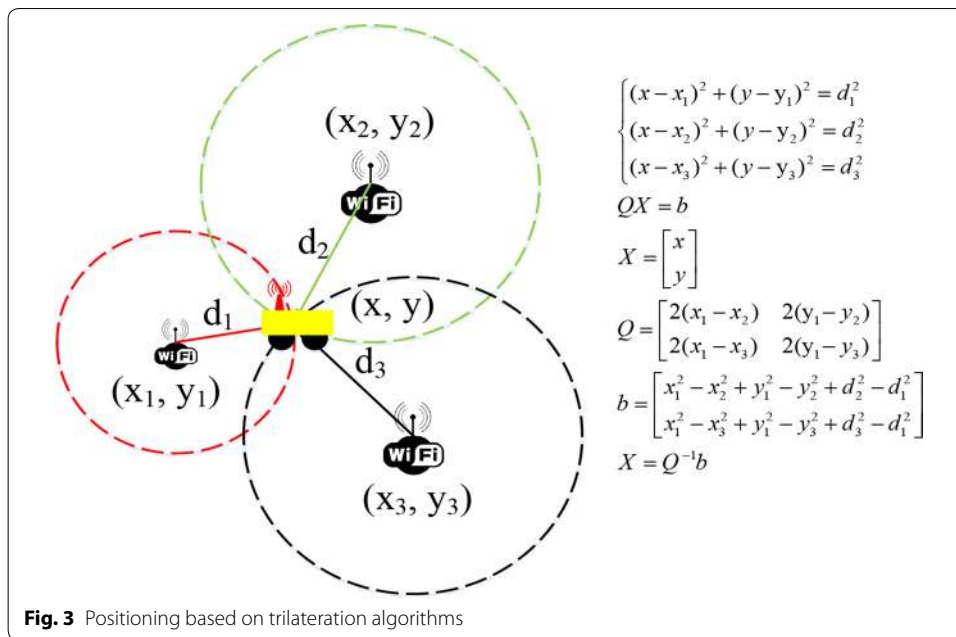
Finally, implementing a cloudlet-based cloud computing system enabling Wi-Fi indoor positioning and navigation is possible for the following reasons. First, because a cloudlet supports resource- and latency-sensitive applications [34], it can provide location-based services such as indoor navigation for people or robots, personnel, asset tracking, guiding blind people, factory automation, workplace safety, locating patients in a hospital, and location-based advertising [1]. Second, a cloudlet provides not only the reference access point locations but also all location information of devices in the network to find the route path for an indoor cart. Moreover, the moving edge cloud in an indoor cart can determine the distances between it and the reference access points using a received



signal strength indication (RSSI)-based method. Similarly, [40], our system can estimate the location of the indoor cart as the location of the reference access point that is located closest to the indoor cart.

The contributions of this paper are as follows:

1. We propose a novel cloudlet-based cloud computing system enabling Wi-Fi indoor positioning and navigation. Moreover, with the rapid growth of Internet of Things (IoT) applications and their deployments on cloud computing, our proposed system brings the cloud closer to IoT devices for providing resource- and latency-sensitive applications.
2. We define a core cloud, cloudlet, and moving edge cloud. The core cloud is used to store all of the object information, such as global position and status, while the cloudlet stores all specific information for the objects. The moving edge cloud is embedded to a task-driven indoor mobile robot, referred here to as a self-driving indoor cart. The moving edge cloud determines the route path and makes movement decisions.
3. We propose a movement decision algorithm for a self-driving cart. A movement decision is made based on measurements of the RSS at a moving edge cloud, which is embedded in the self-driving cart. Consequently, the navigation of the self-driving indoor cart is adjusted in accordance with its current position and the position coordinates of the access points.
4. In real-world indoor environments, such as a one-floor scenario, although our system uses only one access point to estimate the location of an indoor cart, as con-



trusted with standard methods using at least three access points, the experimental results for our system are superior to those of the standard methods in terms of the accuracy of navigation.

- Our proposals were tested using a self-driving indoor cart and real-world indoor environment on the third floor of the Computer Science and Engineering building at Kyung Hee University, Korea.

**Related works**

Wi-Fi survey-based indoor positioning techniques are generally divided into two types, trilateration and triangulation algorithms [17, 43, 44], as shown in Figs. 3 and 4. A trilateration algorithm incorporates ToA, TDoA, RSS, and RToF techniques, while

a triangulation algorithm incorporates AoA and AoD techniques. Trilateration algorithms measure the distance from multiple known access points, while triangulation algorithms compute the angles relative to multiple known access points. A comparison of trilateration and triangulation algorithms is provided in Table 2. The trilateration algorithm calculates the exact location of a user, given the exact location of access points and distances from each access point to the user.

As shown in Table 2, the issues of Wi-Fi-based indoor positioning techniques for use in a real-world indoor environment are as follows.

1. It is difficult to determine a line-of-sight channel between the transmitter and the receiver.
2. A complex radio propagation model must be considered with the multipath effect in indoor environments.
3. Triangulation algorithms show poor accuracy of the estimated location because the multipath affects both the time and the angle of an arrival signal.

A typical indoor self-driving cart consists of four modules: perception, localization, navigation, and motion [45]. One important module known as localization is a key prerequisite for success in navigating the robot, which requires exact identification of current localization [46]. Moreover, the indoor self-driving cart must know which movements to make until it reaches the goal position when the localization is tested. This has led to extensive study of the localization and navigation of mobile robots. We summarize a comparison between our solution and other indoor positioning systems in Table 3.

Cordeiro et al. [47] estimates and records graphically the absolute position of the robot based on odometry. A robot can autonomously move following the desired trajectory while avoiding detected obstacles based on depth images. In addition to odometry, Bessa et al. [48] use computational vision in omnidirectional images to estimate the localization of a robot.

Zhang et al. [49] use QR codes as landmarks to provide global pose references. The QR codes must be placed on the ceiling. A robot uses a camera at the top to read these QR codes more quickly. Thus, the localization of the robot is estimated based on the positional relationship between the camera and the QR codes when QR codes can be recognized. Moreover, a robot uses a laser ranger finder to avoid collisions. The authors also use the Dijkstra algorithm to plan a global path and the dynamic window approach to plan local paths.

Mota et al. [50] estimate the position of a robot in a map represented by the incidence matrix of a Petri net (PN), and the use of PN dynamics as the cognition system by evaluating the use of RFID technology. First, cards with RFID technology were placed at each intersection of the pathways of a structured environment (labyrinth). Second, a robot was equipped with an RFID reader on its bottom. A robot moves until it passes over cards with RFID. At each intersection, the robot performs actions, such as turning right or left according to the map defined in its algorithm. Next, it goes straight to the next card. Moreover, the authors use a black line to connect each card to its neighbor cards. The robot is equipped with three infrared sensors to detect and follow these lines.



**Table 2 Wi-Fi based indoor positioning techniques**

Trilateration algorithms		Triangulation algorithms	
Name	ToA [22–24]	TDoA [25, 28]	AoA [26, 27]/AoD [28]
Method	The distances are directly measured with respect to the propagation time	Examine the difference in time at which the signal arrives at multiple reference access points [17]	Find the intersection of several pairs of angle direction lines [17]
Advantages	It can be deployed with various signals, such as the direct sequence spread-spectrum (DSSS) [22] or ultrawide band (UWB) signal [23]	The cross correlation between the signals received at a pair of reference access points can be used to estimate an accurate location [25]	It does not require time synchronization between reference access points [26, 27]
Drawbacks	Requires respect to signal from at least three reference points A timestamp must be labeled in the transmitting signal [17]	The multiple reference access points must share a precise time reference and reference signals	The angle direction is affected by shadowing or multipath reflections [28]
		An RSS-based method is used to estimate the range of the difference between the transmitted signal strength and the received signal strength [42]	An observational error, which is the difference between a measured quantity and its true value, of only a few meters can be achieved [41]
		Path-loss models will be changed owing to the different indoor environment conditions, such as severe multipath fading and shadowing	It is difficult to model the exact delay or processing time in a real environment



**Table 3 Comparisons of localization and navigation system of indoor mobile robots**

Author	Localization	Cognition	Operation
Corderio et al. [47]	Odometry	Line following	A robot can autonomously move following the desired trajectory while avoiding detected obstacles based on depth images
Bessa et al. [48]	Pattern recognition techniques in omnidirectional images	Artificial neural networks in omnidirectional images	A robot uses pattern recognition techniques in omnidirectional images to estimate the localization of the robot
Zhang et al. [49]	QR code	Path planning is performed using the Dijkstra algorithm and dynamic window approach	QR codes are used as landmarks to provide global pose references for mobile robot localization and navigation Uses a Laser Ranger Finder (LRF) to avoid collisions
Mota et al. [50]	Cards and RFID reader	Line following and dynamics of the Petri nets	A robot is equipped with three infrared sensors to detect and follows a black line connecting each card A robot moves until it passes over cards with RFID At each intersection, a robot performs actions, such as turning right or left according to the map defined in its algorithm. Next, it goes straight to the next card
Our work	A cloudlet-based cloud computing approach	Path planning is made by Dijkstra algorithm and consists of Internet Protocols (IPs) address of APs	A robot is equipped with Raspberry Pi as a wireless access point to connect to APs Cloudlets are deployed at APs A robot moves until it reaches stable segment of AP defined in its path planning At each stable segment of AP, a robot performs actions, such as turning right or left or going straight according to movement decision algorithm

In the mobile edge computing environment, a multi-modal framework for indoor localization tasks was proposed in [51]. In this system, machine learning models are used for processing RSS based indoor localization tasks. Nevertheless, the presence of unstable factors that affect RSS is a major drawback. It will be failed to repeat the same performance in practical situations.

On the other hand, smartphone-based indoor localization and navigation systems rely on RSSI from BLE beacons and inbuilt sensors of smartphones for localizing the user in indoor area, such as Bluetooth receivers, accelerometers, and barometers [52]. Lee et al. [53] use the RSSI of the signal received from Bluetooth beacons to estimate location with the help of the trilateration algorithm. Satan et al. [54] use Bluetooth beacon-based method to trace back the wayfinding. To remove the noise signal, the positioning algorithm [55] estimates the distance between client and beacon, based on Bluetooth RSSI values and log-distance path loss model. In [55], Dijkstra's shortest path algorithm is used to find the shortest route for navigation. Yu et al. [56] propose a dead reckoning algorithm that combines Bluetooth and multiple sensors to improve localization

accuracy. This algorithm includes multisensor-based position estimation, Bluetooth model-based position estimation, and Kalman filter fusion. However, Bluetooth beacon-based indoor localization and navigation systems require for client-based solutions and have a relatively small range (up to 30m). Especially, Sadowski et al. [57] compare between Wi-Fi, BLE, ZigBee, and the long-range wide area network (LoRaWAN) for use in an indoor localization system in terms of localization accuracy and power consumption when IoT devices are used. They proved Wi-Fi to be the most accurate [57].

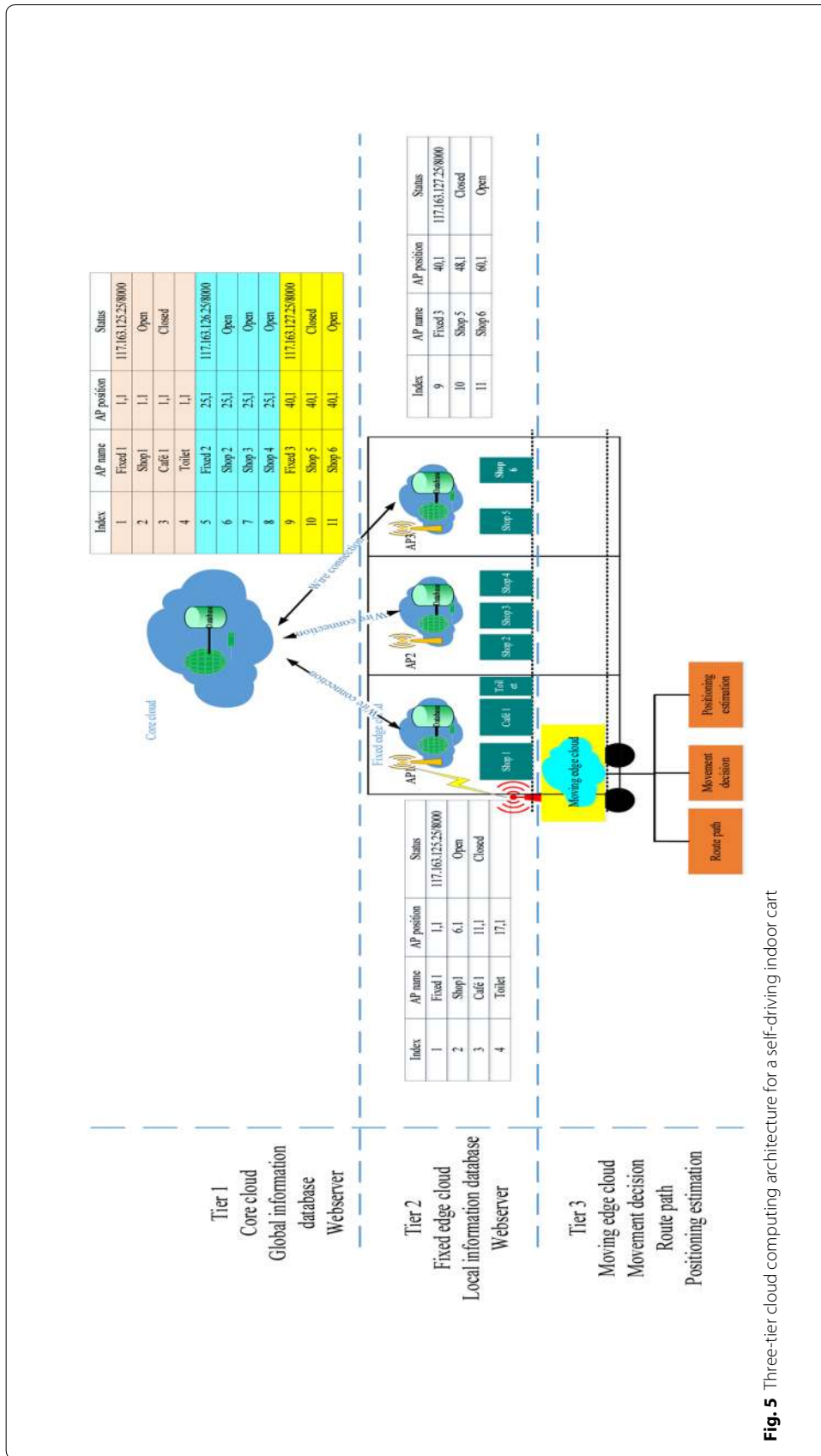
To reduce the negative effects arising from the propagation model, we propose a new system structure that combines a cloudlet-based cloud computing system and indoor positioning and navigation techniques using an RSS-based method. Our system can estimate the position accurately and navigate the task-driven, self-driving indoor cart, as discussed in the next section.

### System structure

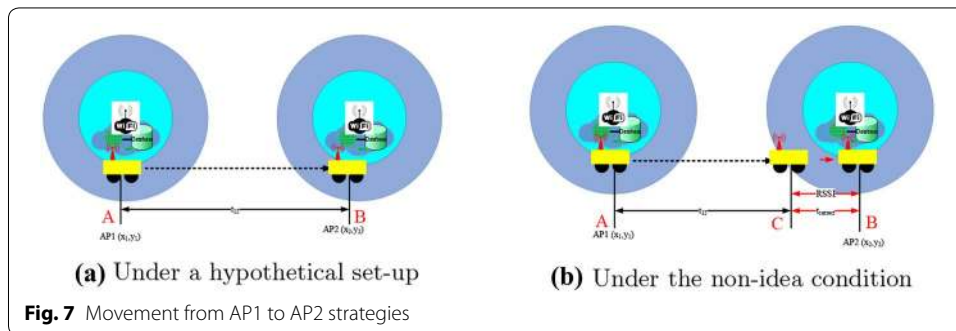
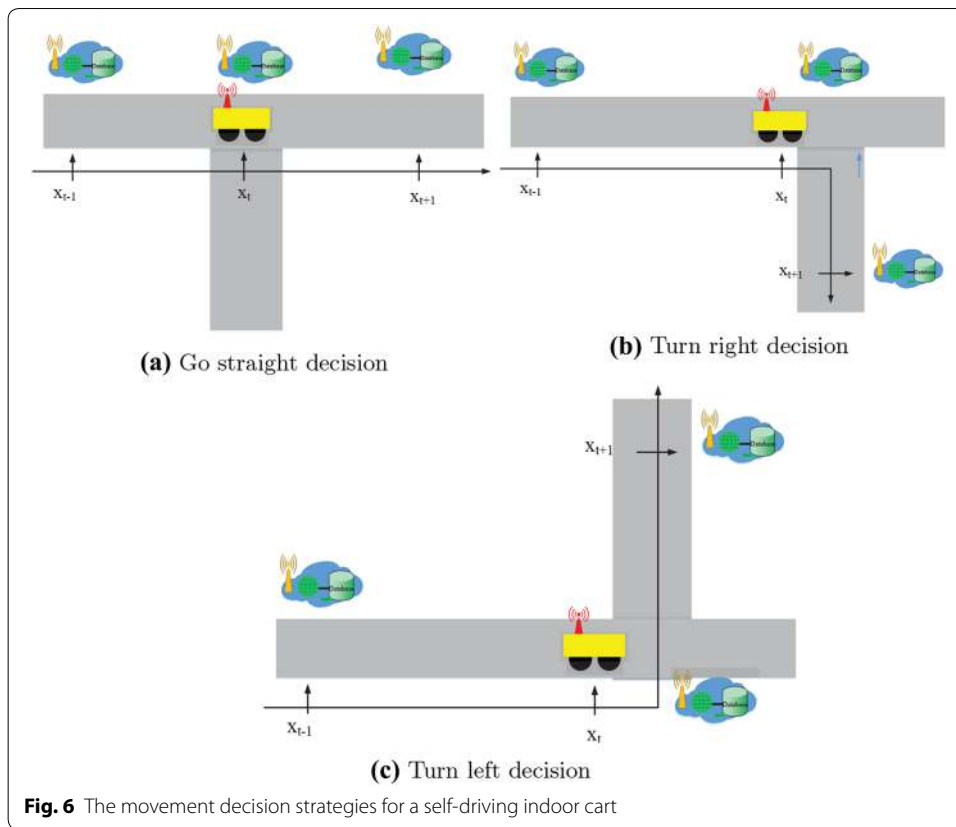
For integration of the Wi-Fi access point with the cloud network, we propose a three-tier architecture for a self-driving indoor cart, as shown in Fig. 5. In the first tier, a core cloud connects with cloudlets that are placed on the corridors. The core cloud stores all information collected from cloudlets. In the second tier, the cloudlets manage local information such as the specific position and status. Finally, the third tier is the moving edge cloud, which is placed in the self-driving indoor cart. Based on the information requested from the cloudlet or core cloud, the moving edge cloud calculates the route path for reaching a destination point. The route path includes the names of the access points (APs) that the cart will be moved past. Note that the route path provides the optimized shortest distance from the starting point to the destination point.

In the moving edge cloud, a movement decision is made when the self-driving indoor cart moves within the AP's coverage. There are three alternatives depending on the route path: go straight (Fig. 6a), turn left (Fig. 6c), or turn right (Fig. 6b). Note that the AP must be placed in corridor intersections to make the movement decision. In real-world indoor environments, such as a one-floor scenario, APs are deployed at corridor ceilings, especially at intersections. The movement decision is based on the previous position, current AP position, and next position, as shown in Fig. 6. Additionally, positioning estimates are used for correcting the movement of the self-driving indoor cart due to avoid the obstacle. For example, the cart moves from position A (3, 4) to position B (10, 4) in the deployed grid map, and it estimates that the movement time is 10 s. However, the cart avoided something in the corridors; hence, 10 s later, it is not at point B (10, 4), and we assume it is at point C. These A, B, and C points are shown in Fig. 7b. The cart has to estimate its current position and move to point B (10, 4) using positioning estimation.

In this paper, we propose a new movement decision algorithm based on comparing the current position  $(x_{t, cart}, y_{t, cart})$  and target position  $(x_{target}, y_{target})$ . As described in Fig. 6, the difference between two values  $x_{t, cart}$  and  $x_{target}$  are used to make go-ahead decision by comparing it with the threshold,  $\delta$ . Additionally, the turn left and right decisions are based on the difference between the  $y$  axis values,  $(y_{t, cart}, y_{target})$  comparing with the threshold,  $\delta$ . The self-driving cart can reach the target position by updating the current position  $(x_{t, cart}, y_{t, cart})$ , as shown in Fig. 8. In the indoor environment, we divide the distance from the beginning point to the destination point into



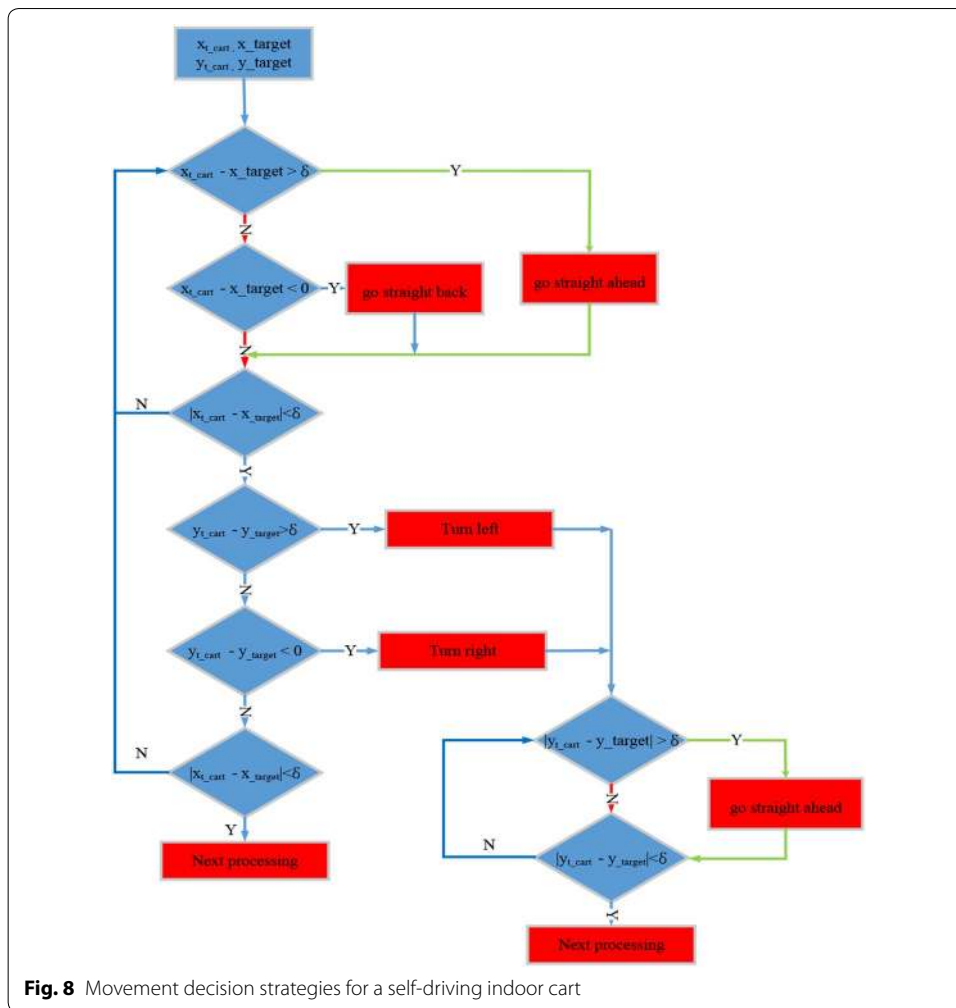
**Fig. 5** Three-tier cloud computing architecture for a self-driving indoor cart



segments according to the respective AP locations. The self-driving cart will move among APs in accordance with movement decisions based on its current position and AP positions (known as target positions). Note that the threshold,  $\delta$ , is determined based on the stable segment of Wi-Fi coverage, as described in the next section.

### Path planning using Dijkstra’s algorithm

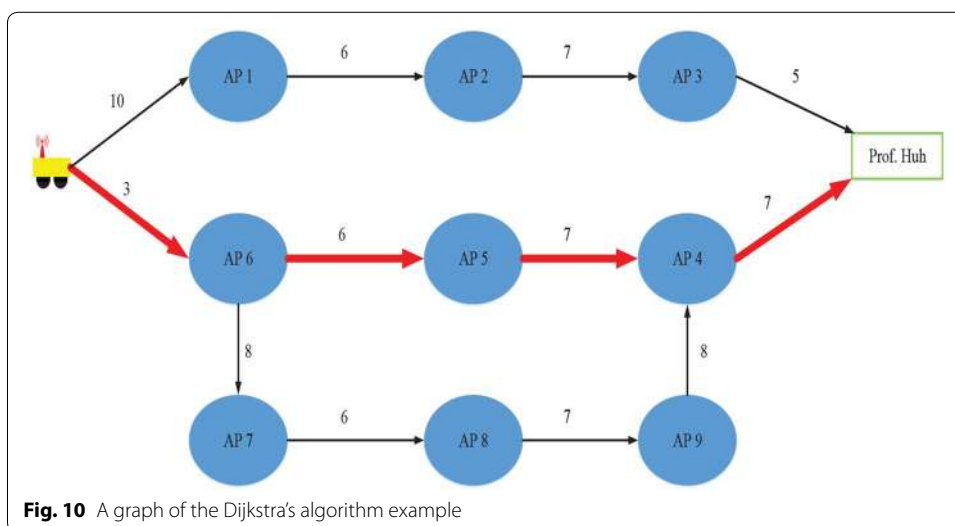
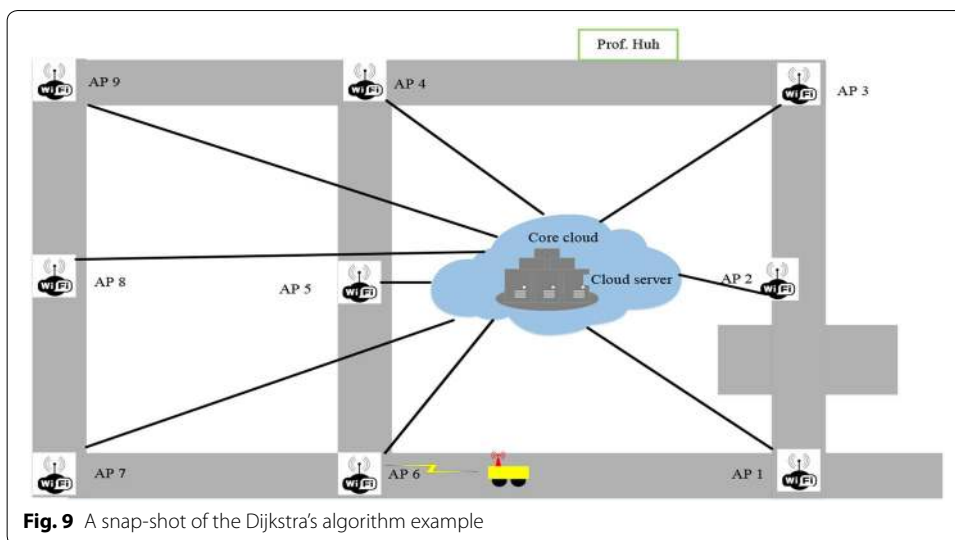
According to [58], Dijkstra’s algorithm is typically used in the indoor environment because it is easy to implement for various environments. Dijkstra’s algorithm is a special form of dynamic programming and a breadth-first-search algorithm for finding the shortest paths from a single source vertex to all other vertices [59]. The



shortest path between two vertices is defined as a path with the shortest length, called link-distance. In Dijkstra’s algorithm, the distance from  $u$  to  $v$ , denoted by  $w(u, v)$ , is the length of a path if there is a path from  $u$  to  $v$ . Otherwise, it is  $\infty$ .

In this paper, the core cloud stores all information of each AP in the network including its position and neighboring AP positions. Note that the core cloud only stores AP’s information if it has a cloudlet. When the indoor cart requests network information to the core cloud, the core cloud will forward all information to it. Based on collected information, the indoor cart will use Dijkstra’s algorithm to find the shortest path between its current position and target position through AP positions. A generalization of Dijkstra’s algorithm can be found in [58]. Dijkstra’s algorithm operation is summarized as follows.

1. Initially, we assign  $\text{Node}(A) = 0$  as the weight of the initial node and  $w(x) = \infty$  to all other nodes, where  $x$  represents the other nodes.
2. Search  $x$  node for which it has the smallest temporary value of  $w(x)$ . If  $w(x) = \infty$  or there are no temporary nodes, the operation will stop the algorithm. The node  $x$  is



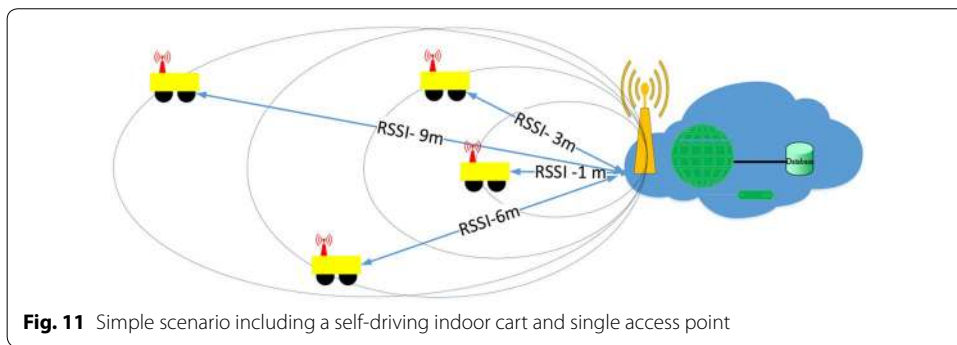
now labeled as permanent and as the current node, meaning parent of  $x$  and  $w(x)$  will stay fixed.

3. For each node adjacent to  $x$  labeled  $y$  which are also temporary, the operation compares and updates as follows. If  $w(x) + W_{xy} < w(y)$ , then  $w(y)$  is updated to  $w(x) + W_{xy}$ , where  $W$  is the cost of the adjacent node. Then, the operation assigns  $y$  to have parent  $x$ .
4. The operation repeats the process from 2 until the shortest path is found.

Figure 9 illustrates the example from the Dijkstra-based scenario. There are APs that include the cloudlet system and their information is stored in a core cloud. The indoor cart requests prof. Huh's room information to the core cloud through the closest AP.

**Table 4 Finding shortest path label from robot to room**

Iterations	Robot	AP1	AP2	AP3	AP4	AP5	AP6	AP7	AP8	AP9	Room
Robot	0	10 <sup>1</sup>	∞	∞	∞	∞	3	∞	∞	∞	∞
{Robot, AP6}	0	10 <sup>1</sup>	∞	∞	∞	9 <sup>6</sup>	<b>3<sup>6</sup></b>	11 <sup>6</sup>	∞	∞	∞
{Robot, AP6, AP5}	0	10 <sup>1</sup>	∞	∞	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	∞	∞	∞
{Robot, AP1}	0	10 <sup>1</sup>	16 <sup>1</sup>	∞	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	∞	∞	∞
{Robot, AP6, AP7}	0	10 <sup>1</sup>	16 <sup>1</sup>	∞	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	17 <sup>6</sup>	∞	∞
{Robot, AP1, AP2}	0	10 <sup>1</sup>	16 <sup>1</sup>	23 <sup>1</sup>	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	17 <sup>6</sup>	∞	∞
{Robot, AP6, AP5, AP4}	0	10 <sup>1</sup>	16 <sup>1</sup>	23 <sup>1</sup>	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	17 <sup>6</sup>	∞	23 <sup>6</sup>
{Robot, AP1, AP2, AP3}	0	10 <sup>1</sup>	16 <sup>1</sup>	23 <sup>1</sup>	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	17 <sup>6</sup>	∞	23 <sup>6</sup> < 28 <sup>1</sup>
{Robot, AP6, AP5, AP4, Room}	0	10 <sup>1</sup>	16 <sup>1</sup>	23 <sup>1</sup>	16 <sup>6</sup>	9 <sup>6</sup>	3 <sup>6</sup>	11 <sup>6</sup>	17 <sup>6</sup>	∞	23 <sup>6</sup>



Once the core cloud receives the requested packet for an indoor cart, it will send network information to the cart. By collecting AP information and prof. Huh’s room position, the indoor cart will draw the graph from its current position to prof. Huh’s room position, as shown in Fig. 10. Using a generalization of Dijkstra’s algorithm, we can find the shortest path label from robot to room according to Table 4 following the above-mentioned Dijkstra’s algorithm operation.

### Measuring principles and cloud-based algorithms

#### Measuring principles

To calculate the distance from the transmitter to the receiver based on the IEEE 802.11 standard, the RSSI is used to measure the relative signal strength, typically measured in decibel milliwatts (dBm) or milliWatts (mW). However, there are no specific formulas to compute accurate distances owing to the impacts of real-world indoor environments and manufacturer implementations [60]. According to [57, 61–65] and using a simple path-loss propagation model [66], the RSSI distance measurement is given as

$$RSSI = -10\gamma \log_{10}(d/d_0) + \Delta \tag{1}$$

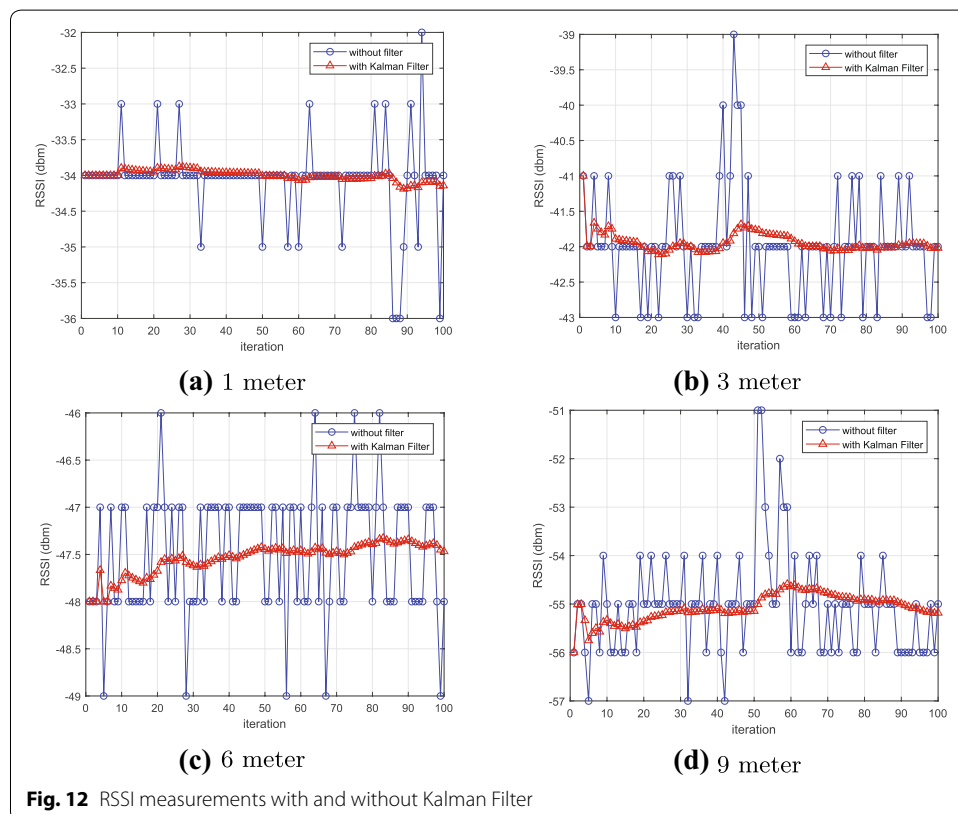
where  $\gamma$  is the path loss exponent depending on the environment,  $d$  is the distance between the mobile device and a reference access point, and  $\Delta$  is a variable accounting for the variation of the mean, often referred to as shadow fading [64, 65, 67]. Moreover,  $d_0$  usually is given as 1 m.



We now consider a single access point and a cart in a simple scenario to measure the RSSI, as shown in Fig. 11. It is clear that environment dependent path loss affects distance measurements in a real environment. One such major issue is that the signal propagation methods in the RSSI-based indoor positioning system are not stable because of the effect of path loss when there are many obstacles, such as the wall, building, weather, or interference. Therefore, many approaches have been proposed to eliminate interference and the effect of path loss, as well as to reduce the positioning error; these techniques include Kalman Filters [68, 69], Extended Kalman Filter-Based Integration [2, 70], weighted K-nearest neighbor [71], and back propagation neural networks optimized by particle swarm optimization [65] algorithms. Herein, we consider a Wi-Fi-based indoor positioning algorithm using the Kalman Filter method, because the location error is controlled within 1.2 m after the Kalman Filter improves the location accuracy of indoor robots effectively [72]. Different RSSI measurements at specific distances are shown in Fig. 12 over 1 m (Fig. 12a), 3 m (Fig. 12b), 6 m (Fig. 12), and 9 m (Fig. 12d) in one closed room.

Based on Fig. 12, we obtain the following conclusions:

1. Without the filter, the RSSI measurements change over time, increasing the positioning error. When the distance increases, the performance of the RSSI measurements is not stable.
2. With the Kalman Filter, the RSSI measurements are stable over a small distance, from 1 to 3 m. At longer distances of 6 and 9 m, the variance is approximately  $-0.5$  dbm.



**Fig. 12** RSSI measurements with and without Kalman Filter

- Alternatively, when we calculate the mean of the RSSI measurements, we find that the average RSSI measurements with and without the Kalman Filter have the same value. Therefore, if a self-driving indoor cart has time to calculate the average RSSI measurement, the positioning error can be reduced. However, in a real environment, this is a challenge since indoor self-driving requires rapid calculation and continuous movement. In the next section, we introduce a new method for position estimation.

**Cloud-based algorithm**

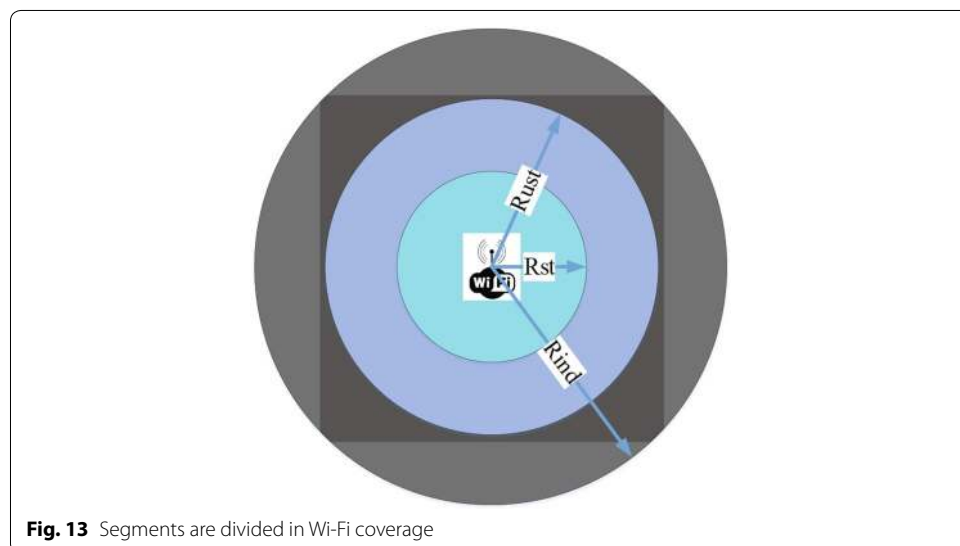
Based on Fig. 12, we divide the Wi-Fi coverage into three segments according to the quality of the RSSI measurement: stable, unstable, and indeterminate measurements. Although the RSSI measurement is affected by interference and path loss, the nearer the mobile device moves to the reference access point, the more stable the RSSI measurement. As shown in Fig. 13, we can examine the circle for a stable segment with  $R_{st}$ , unstable segment with  $R_{ust}$ , and indeterminate segment with  $R_{ind}$ . Note that  $R$  is the radius. The stable segment is defined as in this segment, attaining an RSSI measurement variance of less than 0.5 dBm, while in the unstable segment, the RSSI measurement variance is less than 1.5 dBm. Otherwise, it is called an indeterminate segment.

In the cloud network-based algorithm, APs embedded in the cloud server have known positions. We assume that the speed of the cart is constant,  $v_c$ , and the positions of access points 1 (AP1) and 2 (AP2) are  $(x_1, y_1)$  and  $(x_2, y_2)$ , respectively. The Euclidean distance between AP1 and AP2 is

$$d_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

The time  $t_{12}$ , which moves from AP1 to AP2, is

$$t_{12} = d_{12}/v_c \tag{3}$$



**Fig. 13** Segments are divided in Wi-Fi coverage

Under a hypothetical set-up, as shown in Fig. 7a, the self-driving cart will be in a stable segment of AP2 after  $t_{12}$ . However, in a real indoor environment, the cart has a time delay owing to the need to avoid an obstacle in the corridors. After  $t_{12}$ , we assume that the cart is in the unstable segment of AP2. To move to a stable segment of AP2, the cart must continue moving until the time  $t_{correct}$ . If there are too many obstacles and the cart does not reach even the unstable segment within  $t_{12}$ , the cart will move 1 m until it reaches the stable segment. Note that when the cart arrives at the stable segment, it will run the movement decision algorithm (Fig. 8) with  $(x_{t, cart}, y_{t, cart})$  as the current AP position and  $(x_{target}, y_{target})$  as the next AP position or target position.

The total time  $t_{total}$  to move from AP1 to AP2 is

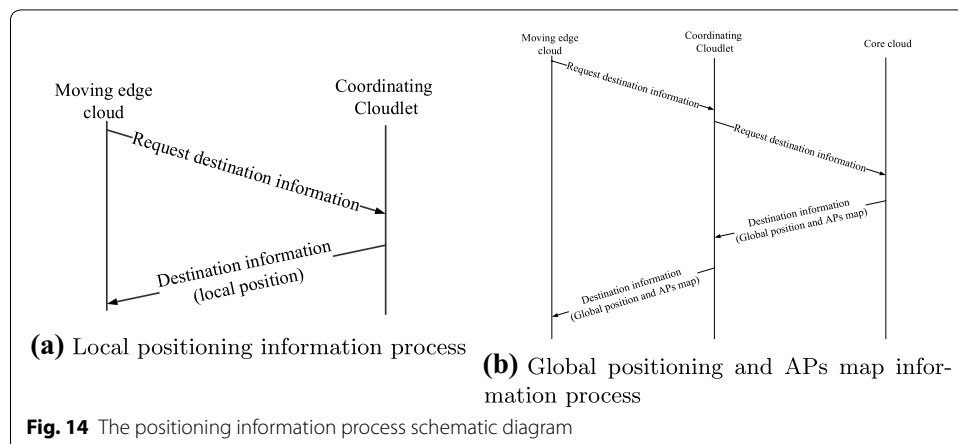
$$t_{total} = t_{12} + t_{correct} + n.1/v_c \tag{4}$$

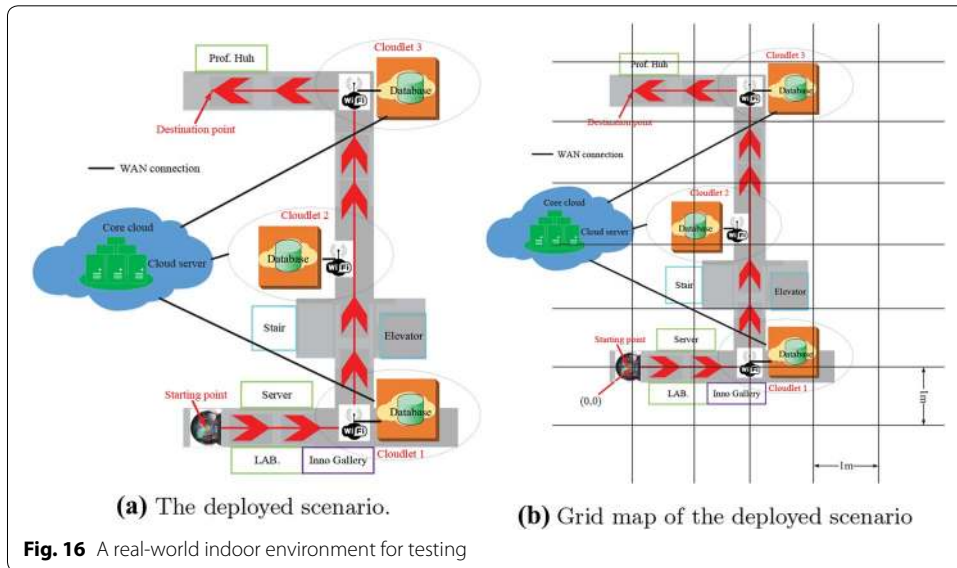
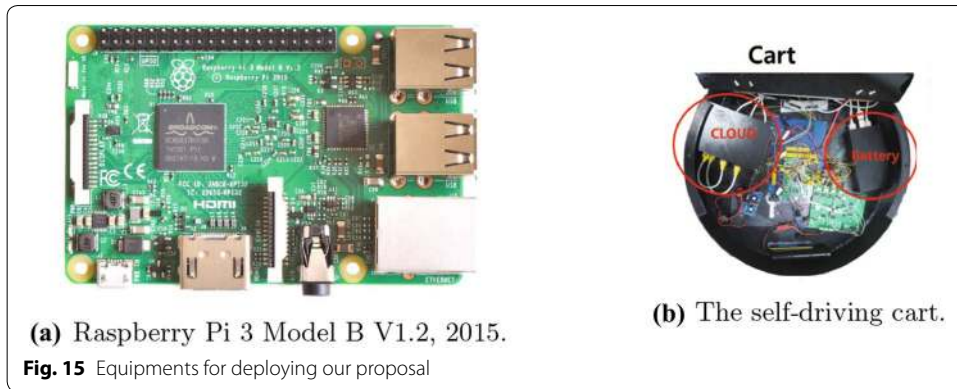
where  $t_{12}$  is calculated according to Eq. (3),  $n$  is number of 1 m increases until the cart reaches the stable segment. The correct time is

$$t_{correct} = d/v_c \tag{5}$$

where  $d$  is determined by Eq. (1) and shown in Fig. 7b.

Finally, as seen in Fig. 5, we implement two processes, a local positioning process, and a global positioning/AP map information process, as shown in Fig 14. Figure 14a shows that the self-driving cart requests the destination’s positioning information from the cloudlet. If there is destination positioning information, the cloudlet forwards this packet to the cart. Note that this destination positioning information is the local position. Otherwise, the cloudlet requests destination positioning information from the core cloud. The core cloud stores all information in the corridors; hence, it forwards the destination positioning packet, including the global positioning/AP map information process, as shown in Fig. 14b. Based on this information, the self-driving cart determines the route path by minimizing the distance between the starting position and the destination point.





### Experimental results

Our indoor localization and navigation system was tested using a Raspberry Pi 3 Model B V1.2, 2015 [73], shown in Fig. 15a, and a self-driving cart, shown in Fig. 15b. We used five Raspberry Pi 3 setups: setups: three for the cloudlets, one for the core cloud, and one for the moving edge cloud. The operation of our implementation is summarized as follows.

- 1 First, we assume that the cart has no information regarding the network, with the target position stored either in the cloudlet (fixed cloud) or the core cloud. At the beginning operation stage, the user chooses a target item for the self-driving cart to reach.
- 2 Second, the moving edge cloud sends the target point to the corresponding cloudlet located closest to that cart. If the cloudlet stores the target item, it feeds back on the

- target location to the self-driving cart. Otherwise, it requests to the core cloud to achieve the target location; then, it sends this target location to the self-driving cart.
- 3 Third, based on the global position and AP map, the cart calculates the route path to minimize the movement time if there is a loop or multiple paths to the same object. The path planning is determined using the Dijkstra algorithm.
  - 4 Four, the cart moves in accordance with the route path. Note that, in the self-driving cart, there are sensors, such as HC-SR04 ultrasonic sensors, used to avoid obstacles.
  - 5 Finally, our proposal is applied to the self-driving cart and it reaches the target item.

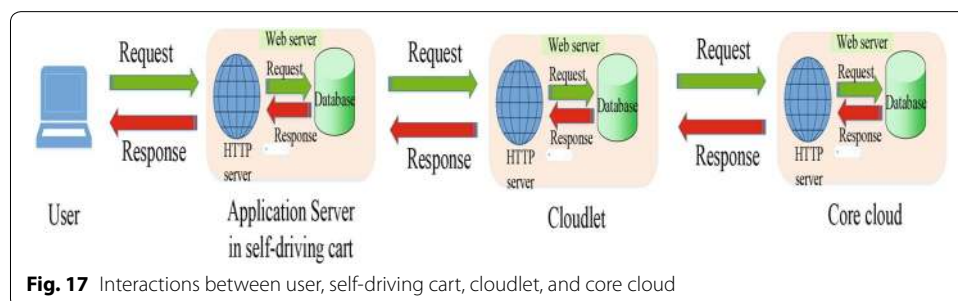
The system was deployed on corridors on the third floor of the Computer Science and Engineering building, Kyung Hee University, Korea. The starting point and destination point are shown in Fig. 16a. We measured and drew the grid map for a real-world indoor environment for testing. Here, both the x and y grid lines are separated by units of 1 m, as shown in Fig. 16b.

To implement our system, we set up a web server, which consists of the database and HTTP server. A Raspberry Pi is used as the web server for the following reason. First, a Raspberry Pi uses an ARM 11 processor running at 700 MHz with 512 MB RAM, and hence is known as a small computer. To learn about web design and server administration for a Raspberry Pi as a web server refer to [74]. It must connect the HC-SR04 ultrasonic sensors and motor controller of the cart with the Raspberry Pi to avoid obstacles and adjust the speed. In addition, we set up a database using MySQL [75]. The interactions between the web servers of our self-driving cart, cloudlet, and core cloud are shown in Fig. 17. In addition, we also set up a Raspberry Pi 3 as an access point [76]. Once the cart moves to that access point coverage, the cart connects with that access point and access webserver if it chooses. Finally, all Raspberry Pi 3 devices are set up as shown in Fig. 18.

A core cloud stores all information from each cloudlet, referred to here as cloudlets #1, #2, and #3. Each cloudlet contains stored local information for each object. Cloudlet #3 contains the local information of Prof. Huh’s room, #301, which is located at  $(-5, 0)_{CI3}$ , whereas cloudlet #3 is located at  $(0, 0)_{CI3}$  in its  $x - y$  axis, as shown in Fig. 19.

**Positioning estimation**

We determined  $R_{st}$ ,  $R_{ust}$ , and  $R_{ind}$  in the real-world indoor environment in accordance to Fig. 13. We considered the distance measurements for fifty iterations under a moving



**Fig. 17** Interactions between user, self-driving cart, cloudlet, and core cloud

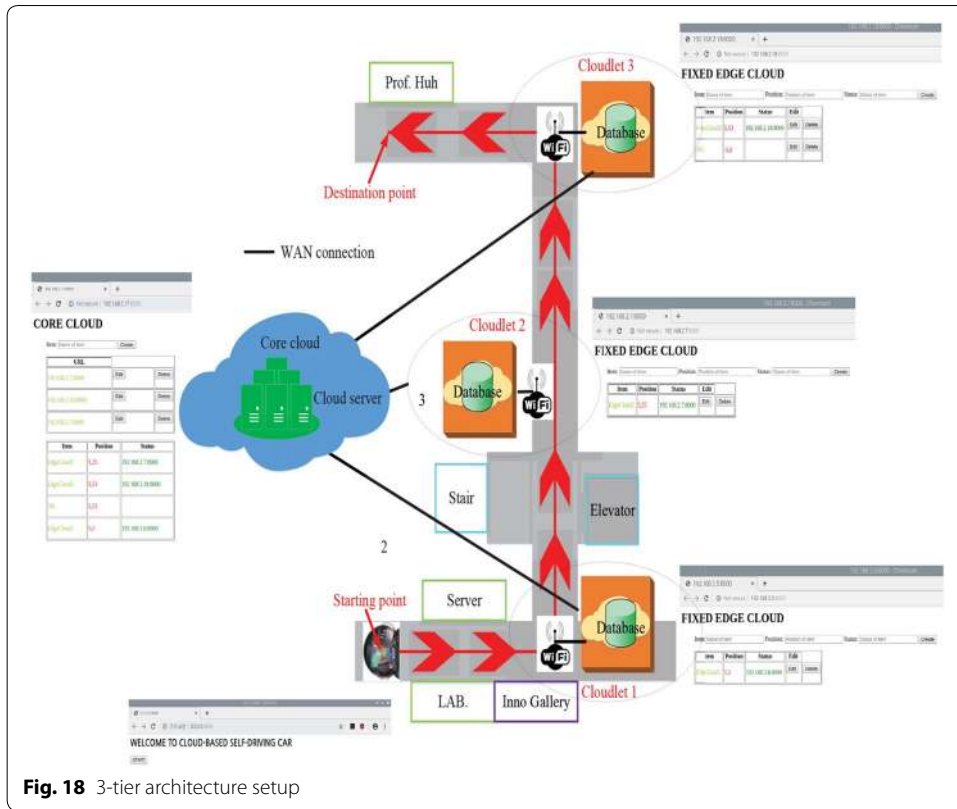


Fig. 18 3-tier architecture setup

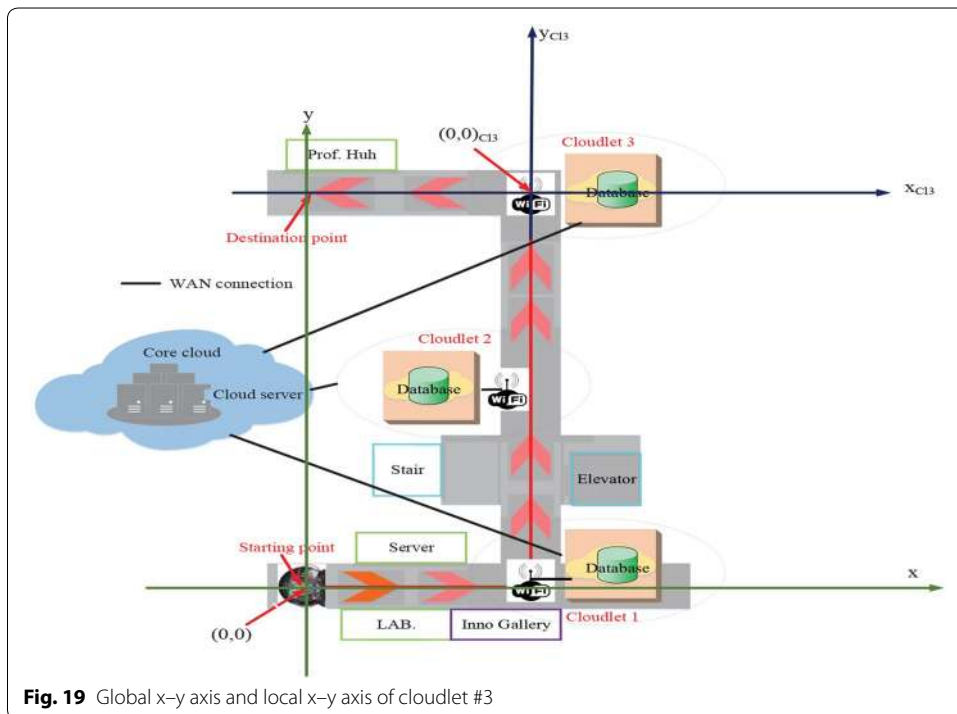
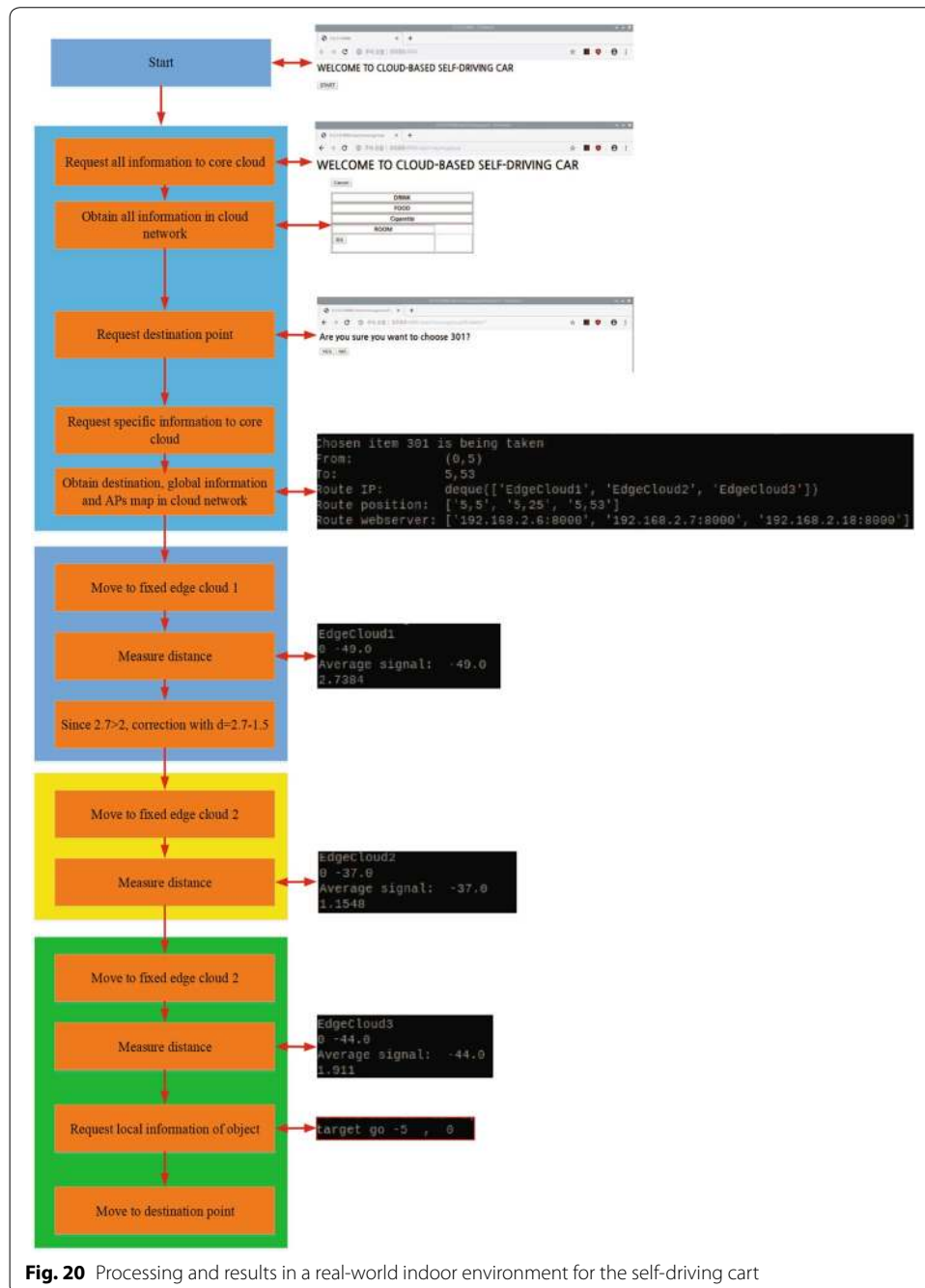


Fig. 19 Global x-y axis and local x-y axis of cloudlet #3

**Table 5 Average of distance measurements using Eq. (1) according to the number of iterations**

Iterations	5 m	4 m	3 m	2 m	1 m
10	2.94	1.91	0.93	2.12	1.78
20	2.37	2.21	1.15	2.05	1.91
30	3.26	2.21	1.24	2.0	1.77
40	3.16	1.91	1.15	2.17	1.71
50	2.37	1.91	1.15	2.07	1.68





edge cloud and one cloudlet by varying the distance from the cart and cloudlet. Observational errors, the differences between measured quantities and their respective true values, are observed. For each iteration, we took the average distance over ten measurements, with the results shown in Table 5. Clearly,  $0 < R_{st} < 2$ ,  $2 < R_{ust} < 3$ , and  $R_{ind} > 4$ .

### Cloud-based algorithm run

We then applied the cloud-based algorithm to the self-driving cart. The processing and results in the real-world indoor environment for the self-driving cart are shown in Fig. 20. Using the cloud network and RSSI-based distance measurements, the cart can move successfully to the destination point during the experiments within the length of the stable segment. We also provide the open-source code at the following links: <https://github.com/icns-distributed-cloud/WebServers> and <https://github.com/icns-distributed-cloud/stm-cartcontroller>. The demo of our proposal also is shown as the following link: <https://youtu.be/h3wYM76PDGs>.

### Conclusion

We proposed a Wi-Fi indoor positioning and navigation method using a cloudlet-based cloud computing system. Our proposal was tested using Raspberry Pi 3 equipment designed with cloudlets, a core cloud, and a self-driving cart, which was equipped with a moving edge cloud. The cloud network not only provides the destination position and wireless access point map but also navigates the cart's movement to reach the destination point. The moving edge cloud calculates the route path, movement decisions, and positioning estimations. The implementation results demonstrated the effectiveness of our proposal. The proposed method can also be used to guide blind people or locate patients in a hospital. In future work, we will enhance our system by decreasing the movement time and total processing time of the indoor cart.

### Abbreviations

SS: Received signal strength; RSSI: Received signal strength indication; GPS: Global positioning system; RFID: Radio-frequency identification; UWB: Ultra-wideband beacon; IMU: Inertial measurement unit; ToA: Time-of arrival; TDoA: Time difference of arrival; AoA: Angle of arrival; AoD: Angle of departure; RTof: Round-trip time-of-flight; DSSS: Direct sequence spread-spectrum; PN: Petri net; AP: Access point; BLE: Bluetooth low energy; LoRaWAN: Long-range wide area network.

### Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00294, Service mobility support distributed cloud technology).

### Authors' contributions

Methodology, Writing—review and editing: TTK and VDN; Resources: TTK; Software: TTK, VDN and X-QP; Supervision: E-NH; Writing—original draft: TTK. All authors read and approved the final manuscript.

### Funding

This study was supported by the the Korea government (No. 2017-0-00294, Service mobility support distributed cloud technology), Korea.

### Availability of data and materials

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

Received: 10 November 2019 Accepted: 15 June 2020

Published online: 16 July 2020

## References

- Li S, Hedley M, Bengston K, Humphrey D, Johnson M, Ni W (2019) Passive localization of standard Wi-Fi devices. *IEEE Syst J*. <https://doi.org/10.1109/JSYST.2019.2903278>
- Cui Y, Zhang Y, Huang Y, Wang Z, Fu H (2019) Novel Wi-Fi/MEMS integrated indoor navigation system based on two-stage EKF. *Micromachines* 10(3):198. <https://doi.org/10.3390/mi10030198>
- Li Y, Zhuang Y, Zhang P, Lan H, Niu X, El-Sheimy N (2017) An improved inertial/Wi-Fi/magnetic fusion structure for indoor navigation. *Inf Fusion* 34(101–119):1566–2535
- Kunhoth J, Karkar A, Al-Maadeed S (2020) Indoor positioning and wayfinding systems: a survey. *Hum Cent Comput Inf Sci* 10:18
- Ali MU, Hur S, Park Y (2019) Wi-Fi-based effortless indoor positioning system Using IoT sensors. *Sensors* 19(7):1496. <https://doi.org/10.3390/s19071496>
- Seol S, Lee EK, Kim W (2017) Indoor mobile object tracking using RFID. *Future Gener Comput Syst* 76:443–451
- Lindo A, García E, Ureña J, del Carmen Pérez M, Hernández Á (2015) Multiband waveform design for an ultrasonic indoor positioning system. *IEEE Sensors J* 15(12):7190–7199
- Zhou C, Yuan J, Liu H, Qiu J (2017) Bluetooth indoor positioning based on RSSI and Kalman filter. *Wirel Pers Commun* 96:4115–4130
- Chen P, Kuang Y, Chen X (2017) A UWB/improved PDR integration algorithm applied to dynamic indoor positioning for pedestrians. *Sensors* 17:2065
- Aykaç M, Ergun E, Noor BA (2017) ZigBee-based indoor localization system with the personal dynamic positioning method and modified particle filter estimation. *Analog Integr Circuits Signal Process* 92(2):263–279
- Wang Ke, Nirmalathas Ampalavanapillai, Lim Christina, Alameh Kamal, Li Hongtao, Skaftidas Efstratios (2017) Indoor infrared optical wireless localization system with background light power estimation capability. *Opt Express* 25:22923–22931 (2017)
- Li X, Zhang P, Guo J, Wang J, Qiu W (2017) A new method for single-epoch ambiguity resolution with indoor pseudolite positioning. *Sensors* 17(4):921
- Shen G, Chen Z, Zhang P, Moscibroda T, Zhang Y, Walkie-Markie (2013) Indoor pathway mapping made easy. In: *Proceedings of the 10th USENIX conference on networked systems design and implementation*, Lombard, IL, USA, 2–5 April 2013. pp 85–98
- Zhou B, Li Q, Mao Q, Tu W, Zhang X, Chen L (2015) ALIMC (2015) Activity landmark-based indoor mapping via crowd sourcing. *IEEE Trans Intell Transp Syst* 16:2774–2785
- Wu C, Yang Z, Liu Y, Xi W (2013) WILL (2013) Wireless indoor localization without site survey. *IEEE Trans Parallel Distrib Syst* 24:839–848
- Gwon Y, Jain R (2004) Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In: *Proceedings of the second international workshop on mobility management & wireless access protocols*, Philadelphia, PA, USA 1 October 2004. pp 2–9
- Liu H, Darabi H, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. *IEEE Trans Syst Man Cybern Part C* 37(6):1067–1080 Nov. 2007
- Cai C, Zheng R, Li J, Zhu L, Pu H, Hu M (2020) Asynchronous Acoustic Localization and Tracking for Mobile Targets. *IEEE Internet Things J* 7(2):830–845
- Hang Wu, Ziliang Mo, Jiajie Tan, Suining He, Gary Chan S-H (2019) Efficient indoor localization based on geomagnetism. *ACM Trans Sens Netw* 15, 4, Article 42 (October 2019), 25 pages
- Liu M, Cheng L, Qian K (2020) Indoor acoustic localization: a survey. *Hum Cent Comput Inf Sci* 10:2
- Beomju Shin, Jung Ho Lee, Taikjin Lee and Hyung Seok Kim (2012) Enhanced weighted K-nearest neighbor algorithm for indoor Wi-Fi positioning systems. In: *2012 8th international conference on computing technology and information management (NCM and ICNIT)*, Seoul, 2012. pp 574–577
- Fang BT (1990) Simple solutions for hyperbolic and related position fixes. *IEEE Trans Aerosp Electron Syst* 26(5):748–753 Sept. 1990
- Peterson BB, Kmiecik C, Hartnett R, Thompson PM, Mendoza J, Nguyen H (1998) Spread spectrum indoor geolocation. *J Inst Navigat* 45(2):97–102
- Correal NS, Kyperountas S, Shi Q, Welborn M (2003) An ultrawideband relative location system. In: *Proc. IEEE conf. ultra wideband syst. technol.*, pp 394–397
- Torrieri D (1984) Statistical theory of passive location systems. *IEEE Trans Aerosp Electron Syst* 20(2):183–197 Mar. 1984
- Van Veen BD, Buckley KM (1988) Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Mag* 5(2):4–24
- Ottersten B, Viberg M, Stoica P, Nehorai A (1993) Exact and large sample ML techniques for parameter estimation and detection in array processing. In: Haykin SS, Litva J, Shepherd TJ (eds) *Radar array processing*. Springer-Verlag, New York, pp 99–151
- Stoica P, Moses RL (1997) *Introduction to spectral analysis*. Prentice-Hall, Englewood Cliffs, p 1997
- Li B, Wang Y, Lee HK, Dempster A, Rizos C (2005) Method for yielding a database of location fingerprints in WLAN. *IEE Proc Commun* 152(5):580–586
- Jekabsons Gints, Kairish Vadim, Zuravlyov Vadim (2011) An analysis of Wi-Fi based indoor positioning accuracy. *Sci J Riga Tech Univ Comput Sci* 44(1):131–137
- Dardari D, Closas P, Djurić PM (2015) Indoor tracking: theory, methods, and technologies. *IEEE Trans Vehic Technol* 64(4):1263–1278
- Turgut Zeynep, Aydin Gulsum Zeynep Gurkas, Sertbas Ahmet (2016) Indoor localization techniques for smart building environment. *Procedia Comput Sci* 83:1176–1181
- Abbas N, Zhang Y, Taherkordi A, Skeie T (2018) Mobile edge computing: a survey. *IEEE Internet Things J* 5(1):450–465
- Satyanarayanan M, Bahl P, Caceres R, Davies M (2009) The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Comput* 8(4):14–23 Oct.–Dec. 2009
- Liu Y, Lee MJ, Zheng Y (2016) Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. *IEEE Trans Mob Comput* 15(10):2398–2410
- Verbelen T, Simoens P, De Turck F, Dhoedt B (2012) Cloudlets: bringing the cloud to the mobile user. In: *Proc. 3rd ACM workshop mobile cloud comput. services (MCS)*, pp 29–36

37. El-Barbary AE-HG, El-Sayed LAA, Aly HH, El-Derini MN (2015) A cloudlet architecture using mobile devices. In: Proc IEEE/ACS 12th int conf comput syst appl (AICCSA), pp 1–8
38. Jararweh Y (2016) SDMEC: Software defined system for mobile edge computing. In: Proc. IEEE int. conf. cloud eng. workshop (IC2EW). Berlin, Germany, pp 88–93
39. Habak K, Ammar M, Harras KA, Zegura E (2015) Femto clouds: leveraging mobile devices to provide cloud service at the edge. In: Proc. IEEE 8th int. conf. cloud comput., New York, NY, USA, pp 9–16
40. Kanaan M, Pahlavan K (2004) A comparison of wireless geolocation algorithms in the indoor environment. In: 2004 IEEE wireless communications and networking conference (IEEE Cat. No.04TH8733), Vol.1, Atlanta, GA, USA, pp 177–182
41. Gunther A, Hoene C (2005) Measuring round trip times to determine the distance between WLAN nodes. In: Proc. netw. 2005., Waterloo, ON, Canada, pp 768–779
42. Zhou J, Chu KM-K, Ng JK-Y (2005) Providing location services within a radio cellular network using ellipse propagation model. In: Proc 19th int. conf. adv. inf. netw. appl., pp 559–564
43. Ugur B, Tenruh M (2014) Increasing RSSI localization accuracy with distance reference anchor in wireless sensor networks. *Acta Polytech Hung* 11(8):103–120
44. Zafari F, Gkelias A, Leung KK (2019) Survey of indoor localization systems and technologies. *IEEE Commun Surv Tutor* 21(3):2568–2599
45. Siegwart R, Nourbakhsh IR (2004) Introduction to autonomous mobile robots. MIT Press, Cambridge, p 2004
46. Lim J, Lee S, Tewolde G, Kwon J (2018) Indoor localization and navigation for a mobile robot equipped with rotating ultrasonic sensors using a smartphone as the robot's brain. In: Proc. IEEE int. conf. electro/inf. technol. (EIT), May 2015, pp 621–625
47. Cordeiro TF (2014) Sistema de detecção e contorno de obstáculos para robótica móvel baseado em sensor Kinect, M.S. thesis, Dept. Ind. Eng., School Ind. Eng., Polytechn. Inst. Bragança, Bragança, Portugal, p 104
48. Bessa JA, Barroso DA, da Rocha Neto AR, de Alexandria AR (2015) Global location of mobile robots using artificial neural networks in omnidirectional images. *IEEE Latin Am Trans* 13(10):34053405–3414
49. Zhang H, Zhang C, Yang W, Chen C-Y (2015) Localization and navigation using QR code for mobile robot in indoor environment. In: Proc. IEEE int. conf. robot. biomimetics (ROBIO), pp 2501–2506
50. Da Mota FAX, Rocha MX, Rodrigues JJPC, De Albuquerque VHC, De Alexandria AR (2018) Localization and navigation for autonomous mobile robots using petri nets in indoor environments. *IEEE Access* 6:31665–31676
51. Li W, Chen Z, Gao X, Liu W, Wang J (2019) Multimodel framework for indoor localization under mobile edge computing environment. *IEEE Internet Things J* 6(3):4844–4853
52. Murata M, Ahmetovic D, Sato D, Takagi H, Kitani KM, Asakawa C (2019) Smartphone-based localization for blind navigation in building-scale indoor environments. *Pervasive Mob Comput* 57:14–32
53. Lee K, Nam Y, Min SD (2018) An indoor localization solution using Bluetooth RSSI and multiple sensors on a smartphone. *Multimed Tools Appl* 77:12635–12654
54. Satan A (2018) Bluetooth-based indoor navigation mobile system, 2018. In: 19th International Carpathian control conference (ICCC). Szilvasvarad, pp 332–337
55. Satan A, Toth Z (2018) Development of Bluetooth based indoor positioning application. In: 2018 IEEE international conference on future IoT technologies (Future IoT), Eger
56. Yu N, Zhan X, Zhao S, Wu Y, Feng R (2018) A precise dead reckoning algorithm based on bluetooth and multiple sensors. *IEEE Internet Things J* 5(1):336–351 Feb. 2018
57. Sadowski S, Spachos P (2018) RSSI-based indoor localization with the Internet of Things. *IEEE Access* 6:30149–30161 2018
58. Yang L, Qi J, Song D, Xiao J, Han J, Xia Y (2016) Survey of robot 3D path planning algorithms. *J Control Sci I Eng* 2016(2016)
59. Fadzli SA, Abdulkadir SI, Makhtar M, Jamal AA (2015) Robotic indoor path planning using Dijkstra's algorithm with multi-layer dictionaries. In: 2015 2nd international conference on information science and security (ICISS), Seoul, pp 1–4
60. Lui G, Gallagher T, Li B, Dempster AG, Rizos C (2011) Differences in RSSI readings made by different Wi-Fi chipsets: a limitation of WLAN localization. In: Proceedings of the 2011 international conference on localization and GNSS (ICL-GNSS), Tampere, Finland, 29–30
61. Golestani A, Petreska N, Wilfert D, Zimmer C (2014) Improving the precision of RSSI-based low-energy localization using path loss exponent estimation. In: Proceedings of the 2014 11th workshop on positioning, navigation and communication (WPNC), Dresden, Germany, 12–13 March, pp 1–6
62. Nowak T, Hartmann M, Zech T, Thielecke J (2016) A path loss and fading model for RSSI-based localization in forested areas. In: proceedings of the 2016 IEEE-APS topical conference on Antennas and Propagation in Wireless Communications (APWC), Cairns, Australia, 19–23 September 2016, pp 110–113
63. Nguyen HA, Guo H, Low KS (2011) Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint. *IEEE Trans Instrum Meas* 60:3619–3628
64. Tuta J, Juric MB (2016) A self-adaptive model-based Wi-Fi indoor localization method. *Sensors* 16:2074 2016
65. Li G, Geng E, Ye Z, Xu Y, Lin J, Pang Y (2018) Indoor Positioning Algorithm Based on the Improved RSSI Distance Model. *Sensors* 18:2820
66. Kumar P, Reddy L, Varma S (2009) Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks. In: 2009 fifth IEEE conference on wireless communication and sensor networks (WCSN), IEEE, New York, pp 1–4
67. Du Y, Yang D, Xiu C (2015) A novel method for constructing a Wi-Fi positioning system with efficient manpower. *Sensors* 15:8358–8381
68. Khalil L, Jung P (2015) Scaled Unscented Kalman Filter for RSSI-based Indoor Positioning and Tracking. In: 2015 9th international conference on next generation mobile applications, services and technologies, Cambridge, pp 132–137
69. Ettliger A, Neuner H, Burgess T (2018) Development of a Kalman filter in the Gauss-Helmert model for reliability analysis in orientation determination with smartphone sensors. *Sensors* 18:414

70. Yim Jaegeol, Park Chansik, Joo Jaehun, Jeong Seunghwan (2008) Extended Kalman filter for wireless LAN based indoor positioning. *Decis Support Syst* 45(4):960–971
71. Wang B, Liu X, Yu B, Jia R, Gan X (2019) An improved Wi-Fi positioning method based on fingerprint clustering and signal weighted Euclidean distance. *Sensors* 19:2300
72. Lu J, Li X (2019) Robot indoor location modeling and simulation based on Kalman filtering. *EURASIP J Wirel Commun Netw*. <https://doi.org/10.1186/s13638-019-1462-9>
73. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Accessed 10 Mar 2019
74. <https://www.raspberrypi.org/blog/piserver/>. Accessed 30 Mar 2019
75. <https://www.mysql.com/downloads/>. Accessed 12th Apr 2019
76. <https://learn.sparkfun.com/tutorials/setting-up-a-raspberry-pi-3-as-an-access-point/all>. Accessed 10th May 2019

### **Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.