IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 September 2022, accepted 18 September 2022, date of publication 21 September 2022, date of current version 8 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208356

APPLIED RESEARCH

Smart Parking Lot Based on Edge Cluster Computing for Full Self-Driving Vehicles

WOOJAE KIM AND INBUM JUNG
Software and System Laboratory, Kangwon National University, Chuncheon 24341, South Korea
Corresponding author: Inbum Jung (ibjung@kangwon.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korea Government (MSIT) under Grant 2019R1F1A1058835.

ABSTRACT One promising area that can be serviced by edge computing in real-time is autonomous driving. Fully self-driving vehicles can operate on roads and in buildings, such as indoor parking lots, using various sensors and communication modules. However, because the communication between indoor parking lots and the outside world is limited, and autonomous vehicles currently lack the real-time performance capabilities needed to process all information independently, it is necessary to develop a control scheme for fully self-driving vehicles in indoor settings. In this study, we propose a smart parking lot for self-driving vehicles based on edge cluster computing. A smart parking lot consists of fixed edges and mobile edge vehicles and uses grid maps for parking lot management. To evaluate the performance of smart parking, we compared parking time and moving distance in existing parking environments. Furthermore, the resource cost and number of data transmissions were analyzed to confirm the number of edges for effective service provision and maintenance.

INDEX TERMS Edge computing, full self-driving, mobile edge, smart parking lot.

I. INTRODUCTION

Edge computing is a scheme where tasks such as event detection, data processing, and storage are performed by edges located at the end of an Internet of Things (IoT) network. In edge computing, edges can process data directly, or collaborate with surrounding edges, without forwarding data to a centralized server such as a cloud. With the edge serving as an interface between the cloud and the end user, the cloud can preserve bandwidth while providing real-time services to the end-user, which improves the quality of service (QoS) [1], [2].

Autonomous driving is a promising area in which edge computing can be used to provide services [3]. Autonomous vehicles are equipped with various modules to handle tasks including communication, calculation, and sensing inside the vehicle. In the case of a fully self-driving vehicle, more processing modules are required for the vehicle to assess various situations without human intervention and respond to changes in the surrounding environment in real-time.

The associate editor coordinating the review of this manuscript and approving it for publication was Young Jin Chun.

However, it is difficult to increase the module density or mount a high-performance module inside a vehicle that offers a limited environment in terms of workload and battery capacity. To solve this problem, studies are underway to apply edge computing to real-time services [4], [5], [6].

Full self-driving vehicles operate both on roads and inside buildings, such as indoor or underground parking lots [7]. In a parking lot, the speed of a vehicle is much slower than on a public road. As mentioned earlier, however, real-time processing is difficult due to the limitations on the vehicle's resources. Therefore, a method for managing the vehicle in the parking lot is required. Various methods are being studied for outdoor parking using in-vehicle global positioning systems (GPS), cameras, and roadside units (RSUs) on the road [8], [9]. However, research on parking in environments where communication with the outside is difficult, such as indoor or underground parking lots, remains insufficient.

In this study, we propose a smart parking lot based on edge cluster computing for self-driving vehicles. The proposed smart parking lot consists of fixed-edge terminals placed in the parking lot itself, and full self-driving vehicles serving as a mobile edge. The physical model of the edge terminal is

Raspberry Pi. Edge terminals and mobile edges deployed in parking lots use a grid map for parking space management, route search, and location estimation. The grid map is managed by edge terminals and is delivered to the mobile edges upon entering the parking lot. The edge terminals are interconnected to form an edge cluster, which communicates with the mobile edge entering the parking lot and provides services such as parking space reservation, release, and guidance.

In a general parking environment, when there are few empty parking spaces, the mobile edge must move until it finds an empty space, and the moving distance increases with the parking lot's size. In a smart parking lot, however, a mobile edge is linked to the system upon entry, enabling it to find a parking space automatically. In this study, we conducted experiments to compare the moving distance and time for a mobile edge to park in a smart parking lot to those in a traditional parking lot environment. We determined the number of effective edge terminals for the smart parking lot service by comparing the energy consumption of the deployed edge terminals for interaction with the number of data transmissions according to the number of edge terminals placed.

The contributions of this paper are as follows:

- Edge cluster computing-based smart parking lot system development and architecture presentation
- Presenting modules for interaction and data processing between mobile edges and edge terminals
- Presenting a parking space searching method of the mobile edge in an indoor parking lot where communication with the outside is difficult
- Presenting edge terminal modules for parking lot management

In the experiment, we used the Raspberry Pi Simulator (RaSim) to test the interaction and effective composition of edges [10]. RaSim can perform edge connection structure control, scheduling, event control, resource management, and mobile edge management using the provided modules. In addition, the virtual edges created by the simulator perform different roles using the role function. The remainder of this paper is organized as follows. Section 2 examines the grid map used in this study and summarizes existing smart parking lot research. The proposed smart parking lot system is described in detail in Section 3. Section 4 describes the experimental setup using RaSim, and Section 5 describes the performance evaluation. Finally, Conclusions are presented in section 6.

II. RELATED WORK

A. GRID MAP IN PARKING LOT

A grid map is a technique for effectively schematizing a space. It is possible to express various obstacles or roads by composing a specific space in the same square grid shape [11]. The grid map has the advantage of being easy to express in a simple form without distinguishing between indoor and outdoor spaces, and is particularly widely used for indoor driving of robots or moving objects. A robot or moving object can use sensors to map its surrounding environment in a grid to estimate its location, even in an environment that it encounters for the first time. Additionally, if the user already knows the indoor structure, a grid map can be created in advance to be used when a moving object estimates its location or moves to a destination. An indoor or underground parking lot is suitable for the creation of a grid map because the entrance, exit, driving direction, and parking space are permanently fixed. Moreover, because the mobile edge is equipped with several sensors and communication modules, it can estimate its location using a grid map and move to its destination.

B. RASPBERRY PI AND EDGE CLUSTER

Raspberry Pi is one of the most used edge for various projects. It is equipped with GB memory, Bluetooth, and Wifi modules. Also, Linux-based Raspberry pi OS is provided, and windows or Ubuntu can be used. In addition to this, various software modules for Raspberry Pi have been released. Edge clusters mean groups consisting of edges such as Raspberry Pi. In general, the edge is low-power-oriented. Therefore, computing resources are limited and Raspberry Pi is difficult to provide a service that needs to process a lot of data independently. Edge cluster groups the surrounding edges into one group and distributes the data to compensate for this shortcoming. Since this method reduces the amount of data processed by each edge, rapid data processing becomes possible.

C. EXISTING SMART PARKING LOT

Smart parking lots have been studied using various methods. In particular, wireless sensor networks (WSNs), the Internet of Things (IoT), and image processing are easy to incorporate into actual parking lots. WSN-based smart parking lots collect data through sensor nodes installed in the parking lot and upload it through sink nodes. Although the basic structure of IoT-based smart parking lots is similar to that of WSN-based ones, they are capable of dynamically changing connection structures, roles of components, and the data processing method to ensure appropriate responses to different events.

The image processing-based smart parking lot, which is currently the most widely used, requires fewer devices than WSNs or the Internet of Things, as a single camera is used to capture and analyze multiple parking spaces. Furthermore, research on vehicle detection through artificial intelligence is also in progress. This chapter describes existing studies related to WSNs, IoT, and image processing, many of which explore various approaches related to the smart parking lot.

1) WSN-BASED SMART PARKING LOT

[12] uses a WSN, web server, and several smartphone applications to enable drivers to find free parking lots. Drivers can check the status of a parking space via applications that are updated through sensors installed in the parking space. The web server is divided into a WSN internal web server and a central web server. The central web server is updated

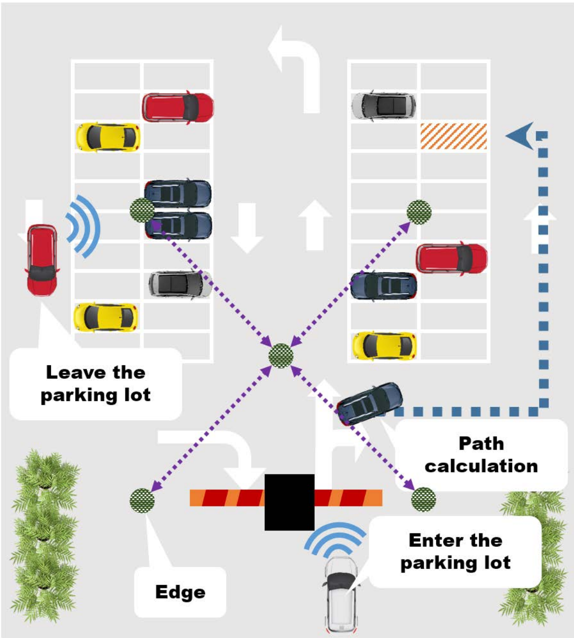

FIGURE 1. Smart parking lot overview.

simultaneously with the internal web server through the driver application.

In [13], the driver was informed of the number and direction of available parking spaces. The system in question also facilitates the monitoring and management of vehicles in parking spaces. This technology is implemented using a WSN, RFID, and ZigBee, and the information obtained from each node is processed using a distributed or centralized method.

In [14], data were collected through the WSN's sensor nodes and then transmitted to the user through the sink node to verify the parking space occupancy status. In this study, data collection, processing, and monitoring were performed using ultrasonic sensors, Arduino, and an LCD, respectively. The parking management process initiates when a car enters the parking lot.

2) IoT-BASED SMART PARKING LOT

In [15], the presence or absence of a vehicle in a parking space was checked using a sensor module called SPIN-V, which consists of a Raspberry Pi, ultrasonic sensor, camera, LED, buzzer, and battery. In this study, the driver was able to reserve and control a parking space through a smartphone app, and the parking lot owner managed the entire space through a monitoring center. Furthermore, the study illustrates the effectiveness of a smart parking system through three scenarios. ('Reserve and park a parking space,' 'Park someone else in the reserved parking space,' and 'The driver does not have an app and parks in an unreserved space')

In [16], the presence of a vehicle in the parking space was detected using infrared and ultrasonic sensors, and the LED lighting was controlled according to the detection result. The system then updated the parking lot status by sending the detected data to the edge and cloud servers through a software model consisting of sensing, network, cognitive, and application layers, thereby reducing the parking lot management costs and parking time.

In [17], an ultrasonic sensor and a long-range (LoRa)-based transmitter module were used to transmit the current parking lot status to a LoRa receiver. The receiver then sent data via Wi-Fi to the user's smartphone app and the IBM Watson cloud. Users could check parking space information through the LoRa receiver or the cloud [17].

3) IMAGE PROCESSING-BASED SMART PARKING LOT

In [18], an intelligent parking guidance and information system was designed using an ARM9 microcontroller. A camera was connected to the microcontroller to capture the parking area and identify empty parking spaces. In the captured image, occupied parking spaces were denoted by red boxed. Upon completion of processing, the image was sent to the user via SMS (Short Message Service). The user could then reserve a parking space by selecting one of the remaining areas.

Reference [19] describes a method for distinguishing occupied and unoccupied spaces among an entire parking space and outputting them on an LCD as the parking lot is filmed in real time. Each image frame in this study was converted to grayscale to assess the state of its respective parking space. Area outside of the parking space was deleted from the converted image. Subsequently, the status of the parking space was determined through binarization, and the occupied and unoccupied spaces were output to the LCD.

In [20], the parking space status was assessed using a Raspberry Pi 3 camera, image processing, and Firebase. As the camera captured the parking lot, the images were processed using the Haar cascade algorithm on Raspberry Pi. As each image was processed, several features were selected using the Haar cascade algorithm. These features were then trained with AdaBoost, and objects were detected using a classifier. Subsequently, the current parking lot status was uploaded to Firebase, which could be accessed remotely to check the parking lot status.

III. SMART PARKING LOT FOR FULL SELF-DRIVING VEHICLES

Figure 1 shows the mobile edge entering a smart parking lot. Upon entry, the mobile edge communicates with the nearest edge to find a parking space, transmitting vehicle information in the process. The edge then maps the vehicle information to an empty parking space, and sends the corresponding parking space information and a grid map to the mobile edge. The mobile edge calculates the route and moves to the empty parking space. Upon leaving, the mobile edge notifies the edge of its departure. The edge then releases the parking space.

A. SMART PARKING LOT DESIGN

Figure 2 shows the smart parking lot system structure. The parking lot operates through the interaction of the edge

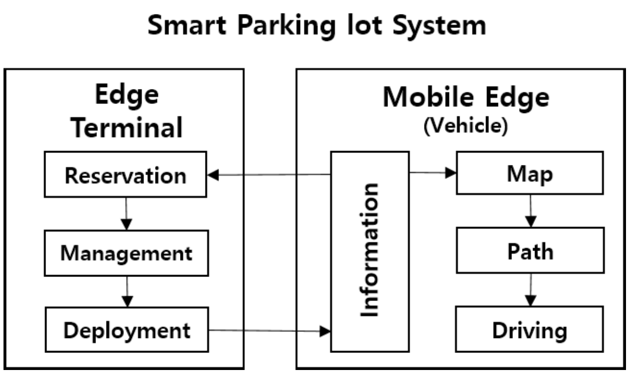

FIGURE 2. Smart parking lot system structure.

deployed in the parking lot and the mobile edge entering and exiting the parking lot.

The edge terminal consists of three modules for vehicle management and a grid map representing the parking lot. The reservation module initiates the parking space reservation process by receiving vehicle information from the mobile edge upon entry. If there are available spaces in the parking area managed by the edge terminal, the received vehicle information is stored and transmitted to the management module, which selects an area. The vehicle information is then mapped to the parking space. Next, mapping information is sent to the deployment module, which sends the parking space information and a grid map to the mobile edge. The mobile edge notifies the edge terminal when it leaves the smart parking lot. The edge then deletes the mobile edge information from the reservation module using the received data. Finally, the management and deployment modules use the data received from the reservation module to convert the reserved space into an empty space.

The mobile edge consists of four modules. The information module stores and transmits vehicle information. The mobile edge that enters the smart parking lot is connected to the edge terminal for parking. It then passes identification information to the reservation module to request parking space. When reservation is complete, the information module receives parking space information and a grid map, and sends it to the map module, which stores the map and sends the information to the path module. The path module then uses the map and information to navigate routes from the current location to a reserved location. The shortest of the navigated routes is sent to the driving module. Finally, the driving module is used to drive to the reserved space via the route received from the path module.

B. INTERACTION BETWEEN MOBILE EDGE AND EDGE TERMINAL

Figure 3 is a sequence diagram of a mobile edge requesting a parking space from an edge terminal. In Figure 3(a), the mobile edge finds a parking space. If the mobile edge finds an edge terminal before it completes parking, they connect to each other. The mobile edge requests a parking space from the connected edge terminal, and the edge terminal checks if there is an empty space in the area it manages. If there is an empty space, the edge terminal informs the mobile edge of the empty space information. The mobile edge moves to the location guided by the edge terminal.

Figure 3(b) shows the situation where there is no empty space in the area managed by the edge terminal. Edge Terminal 1 receives a request for a parking space from the mobile edge. Edge terminal 1 requests a parking space from edge terminal 2 because there is no parking space to guide. Edge terminal 2 returns the location of empty parking space among the areas it manages to edge terminal 1. Edge terminal 1 delivers the parking space location received from edge terminal 2 to mobile edge.

Figure 3(b) can be extended to a case where many edge terminals are connected. If many edge terminals are connected, the mobile edge can receive parking space guidance more quickly. This is because even if there is no empty parking space in the area managed by the edge terminal, it can request from other connected edge terminals. In other words, the mobile edge has the same effect as exploring more parking areas with one request to the edge terminal.

C. MODULES FOR MOBILE EDGE MOVING AND PARKING

The key events that occur in a smart parking lot are parking and leaving. Figure 4 illustrates parking and leaving procedures in (a) and (b), respectively. In Figure 4(a), the mobile edge requests a parking space from the edge which processes it through the reservation module. The reservation module checks the parking request and sends vehicle information to the management module through the addition module. The management module stores the vehicle information in the vehicle management pool, maps it to one of its managed parking spaces, and sends the mapping information to the deployment module.

In the leaving procedure shown in Figure 4(b), the mobile edge sends a departure request with its corresponding vehicle information to the edge. The edge then checks the request and sends the departure information to the management module through the deletion module. The management module deletes the information mapped in the vehicle management pool and sends the remaining information to the deployment module, which then updates the grid map.

Figure 5 illustrates the mobile edge's operation in the smart parking lot. The mobile edge receives a grid map and parking space information through the receiver module. The data received by the receiver module are sent to the map module, which separates it into parking space information and grid map via the separation module. The grid map is sent to and stored by the grid and direction map modules, whereas the parking space information is sent to the navigation module in the path module. The navigation module communicates with the grid and direction map modules and navigates multiple routes. Navigated routes are stored in the path-storage module, and the shortest path is sent to the minimum path module in the driving module. Finally, the minimum path module

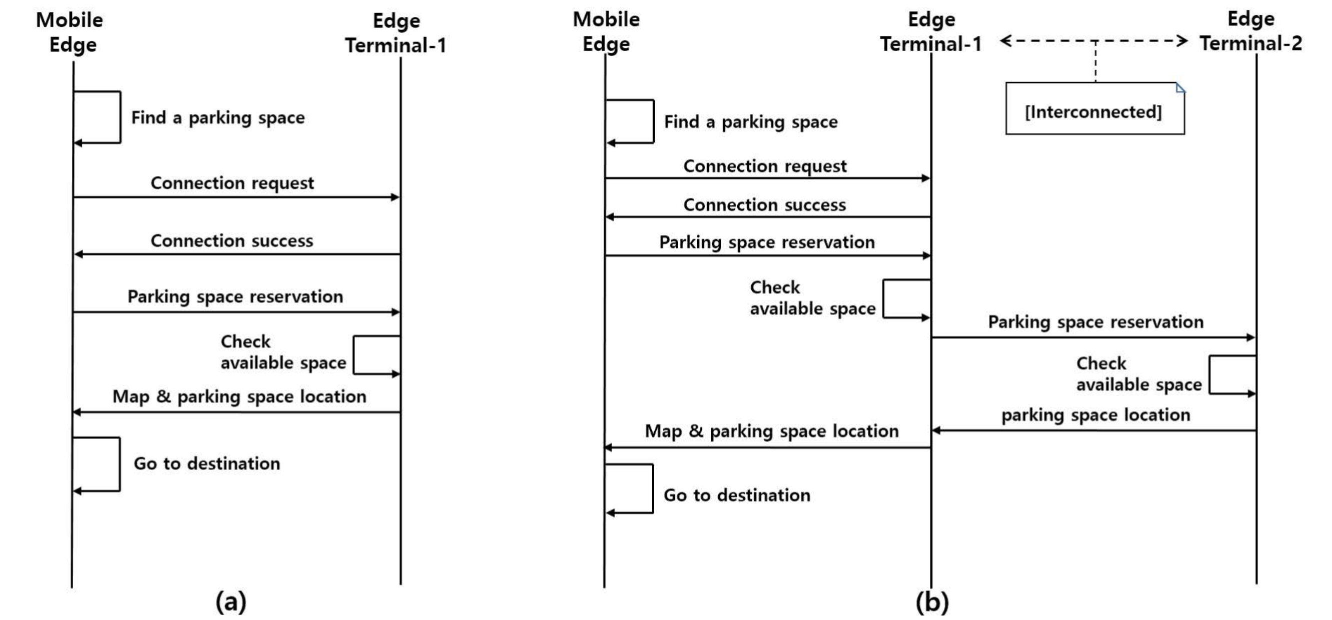

FIGURE 3. Sequence diagram for requesting a parking space.

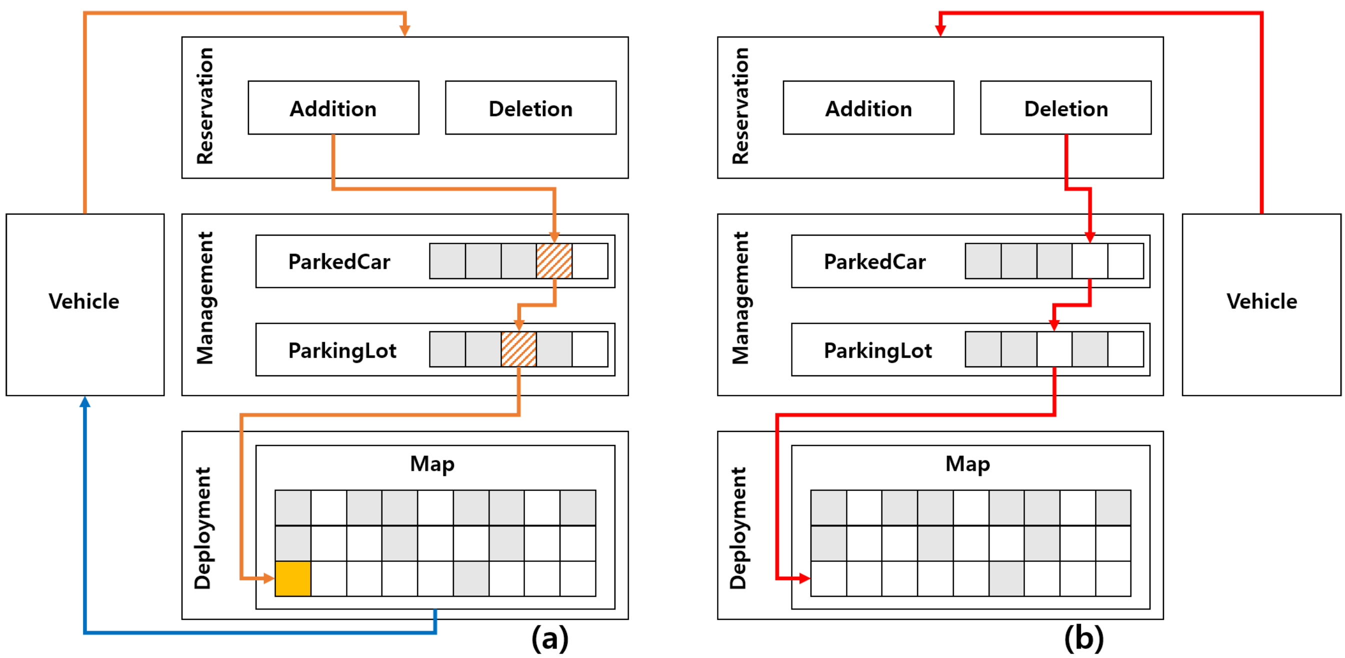

FIGURE 4. Data flow of edge terminal.

sends the path data to the vehicle control unit module, which is then driven to the reserved parking space.

D. ROLE CHANGE THROUGH INTERACTION OF EDGE TERMINALS

Edges are interconnected, and form edge clusters. An edge belonging to a cluster can request a parking space through a connected edge if there are no available spaces in the parking area it manages. When all edges are connected in a smart parking lot, the mobile edge effectively connects to the entire parking lot with one query. Thus, the edge cluster in a smart parking lot reduces the total distance traveled by a mobile edge. In this study, the edges distinguish between followers and spanners. Follower refers to a state connected to another edge, and spanner refers to a state connected to two or more edges.

Figure 6 shows the changes in the role of the edge. The basic role of all edges is that of the follower. Figure 6(a) shows

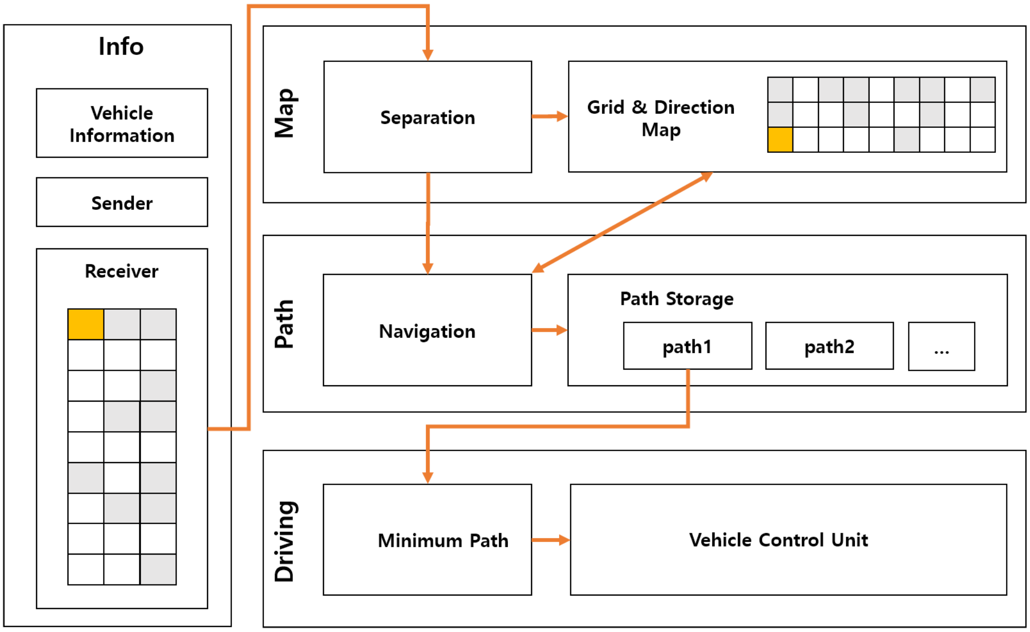

FIGURE 5. Data flow of mobile edge.

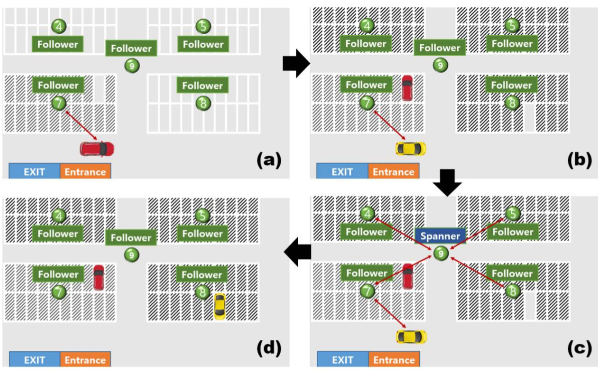

FIGURE 6. Role change of edge terminal.

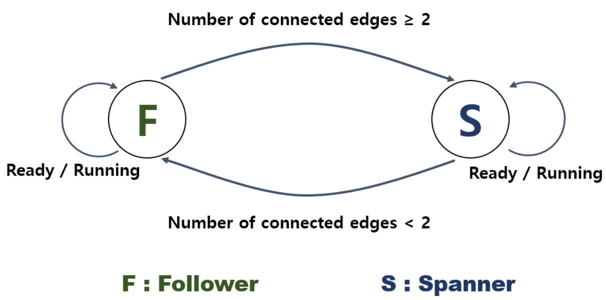

FIGURE 7. State transition of edge terminal.

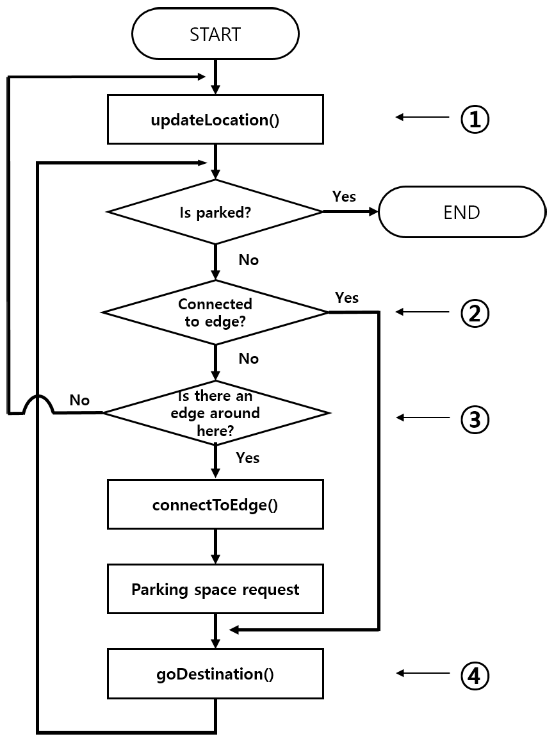

FIGURE 8. Mobile edge operation flowchart.

the mobile edge communicating with Edge 7, which provides it with an empty parking space. Figure 6(b) shows the scarcity of parking space. Although there is an empty parking space managed by Edges 5 and 8, the mobile edge cannot communicate with these edges directly. Therefore, Edge 7 connects with Edge 9 and requests an empty parking space as in Figure 6(c). Edge 9 then connects to Edges 4, 5, and 8 to forward the request to the connected edge. Edge 7 receives information from Edge 8 and then forwards it to the mobile edge. Finally, the mobile edge successfully parks using the received information and the inner modules as in Figure 6(d).

Figure 7 shows a state transition diagram of the change in the role of Edge 9 in Figure 6. This change is determined by the number of connected edges. Edge 9 in Figures 6(a)

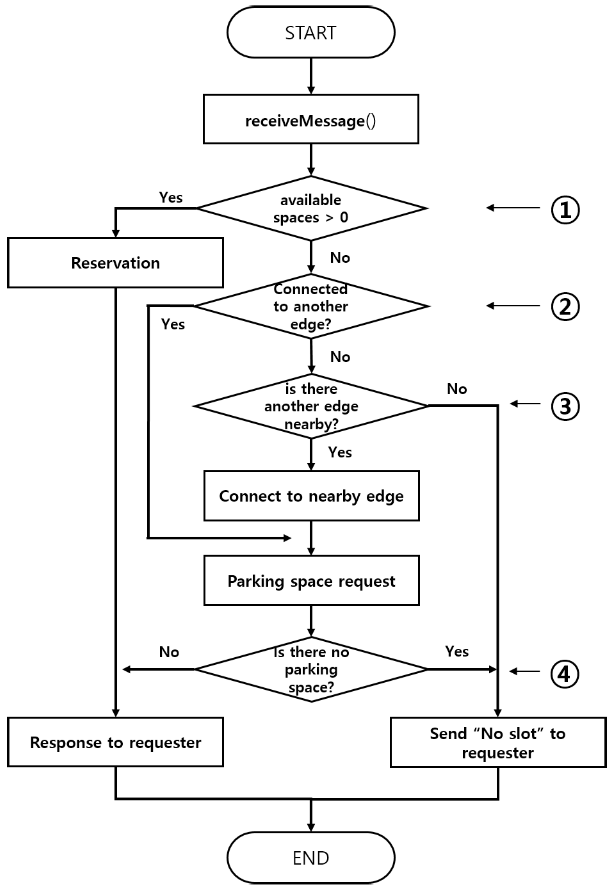

FIGURE 9. Follower edge operation flowchart.

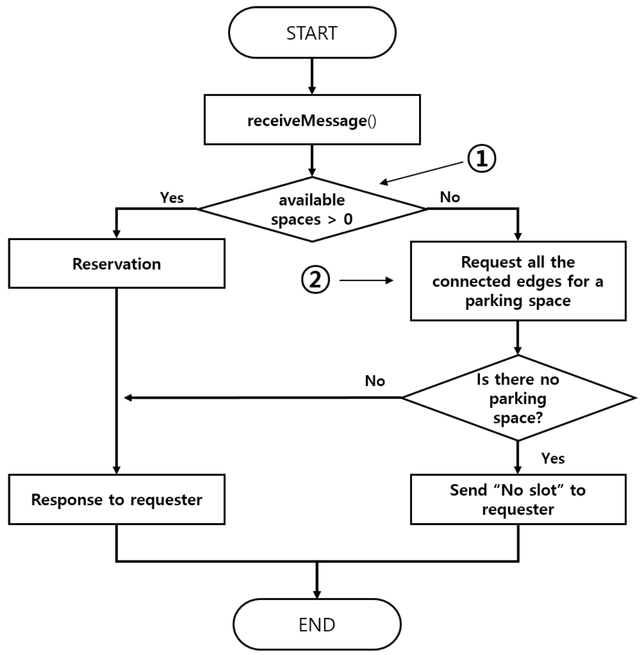

FIGURE 10. Spanner edge operation flowchart.

and 6(b) maintains the role of a follower, because its number of connected edges is zero. However, because there are four connected edges in Figure 6(c), Edge 9 becomes a spanner. All edges of the smart parking lot can switch roles based on the conditions in Figure 7.

E. MOBILE EDGE AND EDGE TERMINAL OPERATION

1) MOBILE EDGE OPERATION

Figure 8 shows the operation of the mobile edge. Upon entry, the mobile edge explores the parking space through updateLocation(), as in ①. The operation is terminated when a parking space is found through updateLocation() without being connected to the edge. If there is no parking space, the mobile edge checks a currently connected edge as in ②. If there are no parking spaces found among the connected edges, the mobile edge checks searches for connectable edges, as in ③. Upon finding a connectable edge, the mobile edge connects to the edge to request parking space. When parking space data are received through the edge, the mobile edge moves to its destination, as in ④.

2) EDGE TERMINAL OPERATION BY ROLES

Figure 9 shows the operational flow when the edge is a follower. When a follower receives a request for parking space, it searches for an available space in its area, as in ①. If an empty space is present, the follower reserves the space and responds to the requester. If there is no empty space, however, the follower checks whether it is connected to another edge, as in ②. If not, the follower connects to a connectable edge, as in ③. In the absence of connectable edges, the follower informs the mobile edge that there are no available spaces. If there is another edge, the follower connects to the edge and requests a parking space. In ④, the follower sends the parking space information to the requester if an available space was found. Otherwise, the follower sends a message to the requester indicating that there are no available spaces.

Figure 10 shows the operational flow when the edge is a spanner. Because a spanner is already connected to two or more edges, there is no connection procedure. Upon receiving a request for parking space, the spanner searches for available spaces in the area managed by itself, as in ①. If an empty space is found, the spanner reserves that space and responds to the requester. Otherwise, as in ②, the spanner requests an empty space for all connected edges. Upon receiving a response, the spanner forwards it to the requester.

This chapter described the design and components of the smart parking lot, mobile edge, and edge terminal. In particular, we focused on the detailed functions and interactions of the mobile edge and edge terminals. The parking lot is efficiently managed through the interaction between the mobile edge and the edge terminal and the interaction between the edge terminals.

IV. IMPLEMENTATION

A. SMART PARKING LOT MODEL AND LAYOUT

This chapter describes the layout for smart parking lot implementation and simulator settings for experimentation. Since the shape of the parking lot is very diverse, one parking

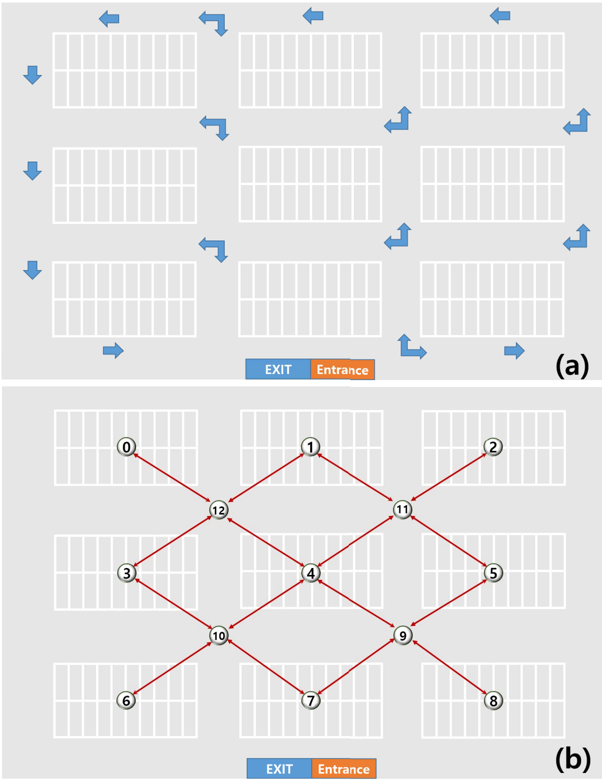

FIGURE 11. (a) Smart parking lot and (b) edge layout.

lot type was set for a clear experiment. Also, we explain how to set the mobile edge and edge terminal options in the simulator.

Figure 11(a) shows the parking spaces and possible movement directions. All mobile edges move in one way and park if there is an empty space. The moving direction at crossroads is randomly determined. Figure 11(b) shows the edge locations. Moving mobile edges requires empty parking space if there is a communicable edge. The edge guides one of its parking spaces.

Figure 11(b) shows a situation where 13 edges are placed. The arrows on each edge indicate connectivity. Figure 12 illustrates the procedure for edge connection using information from Figure 11(b). Because the edges in Figure 12(b-1) are outside the communication range, there is no connection. Due to the placement of Edge 9 in Figure 12(b-2), Edges 4, 5, 7, and 8 can be interconnected because they are located within the communication range of Edge 9. Consequently, a mobile edge that is connected to either of the aforementioned edges can explore all parking spaces managed by the connected edges. In Figures 12(b-3) and 12(b-4), edges 10 and 11 connect all connectable edges. Finally, if 13 edges are placed as in Figure 12(b-5), the mobile edge can explore empty spaces everywhere in the parking lot.

B. SIMULATOR SETTING

We used Raspberry Pi as the physical model of edge and RaSim (Raspberry Pi Simulator) for the experiment. Edges

TABLE 1. Experimental parameters.

Parameter	Value
Size of the smart parking lot (width * height)	70m * 50m
Number of parking space	180
Size of parking space (width * height)	2m * 6m
Speed (km/h)	10
Communication range	16m
Communication type	Bluetooth
Number of edges	0 ∼ 13
Number of parking spaces managed by a single edge (maximum)	20

created in RaSim are set as mobile edges or edge terminals. In the case of edge terminals, the roles can be changed to follower and spanner, so the functions shown in Figures 9 and 10 are implemented and registered as functions.

In this study, RaSim was used to simulate the interaction between the edge and the mobile edge for the purposes of implementation and experiments. RaSim can create virtual things that operate according to their role, which in turn can be changed according to the thing's status. Therefore, RaSim is a suitable simulator for smart parking lots.

Figure 13 shows the RaSim parameter setting screen. In our experiment, RaSim needed to change the value of 'NUMBER_OF_THINGS' to adjust the number of edges. In Figure 13, 'NUMBER_OF_THINGS' denotes the total number of things and 'NUMBER_OF_VEHICLES' denotes the number of mobile edges. Therefore, the number of edges is 'NUMBER_OF_THINGS' - 'NUMBER_OF_VEHICLES,' and the total value changes from 180 to 193.

V. PERFORMANCE EVALUATION

A. EXPERIMENTAL METHOD

Table 1 lists the parameters used in the experiments. There were 180 mobile edges, and the experiment terminated when all mobile edges were parked. All edge communication was facilitated via Bluetooth. The communication range radius was 16 m. The number of edges ranged from 0 to 13, and the edges could connect within the communication range.

A smart parking lot is characterized by at least one edge. In this study, the following three steps were performed to evaluate a smart parking lot.

1) MOVING DISTANCE AND TIME OF MOBILE EDGE

This experiment compared the moving distance and time in smart and general parking lots. In particular, when few available spaces were left, the existence of edges in the parking lot has a significant impact on the moving distance and time of the mobile edge. This was verified in this experiment.

2) NUMBER OF TRANSMISSIONS ACCORDING TO THE NUMBER OF EDGES

In this experiment, we compared the number of transmissions based on the number of edges. The experimental results can

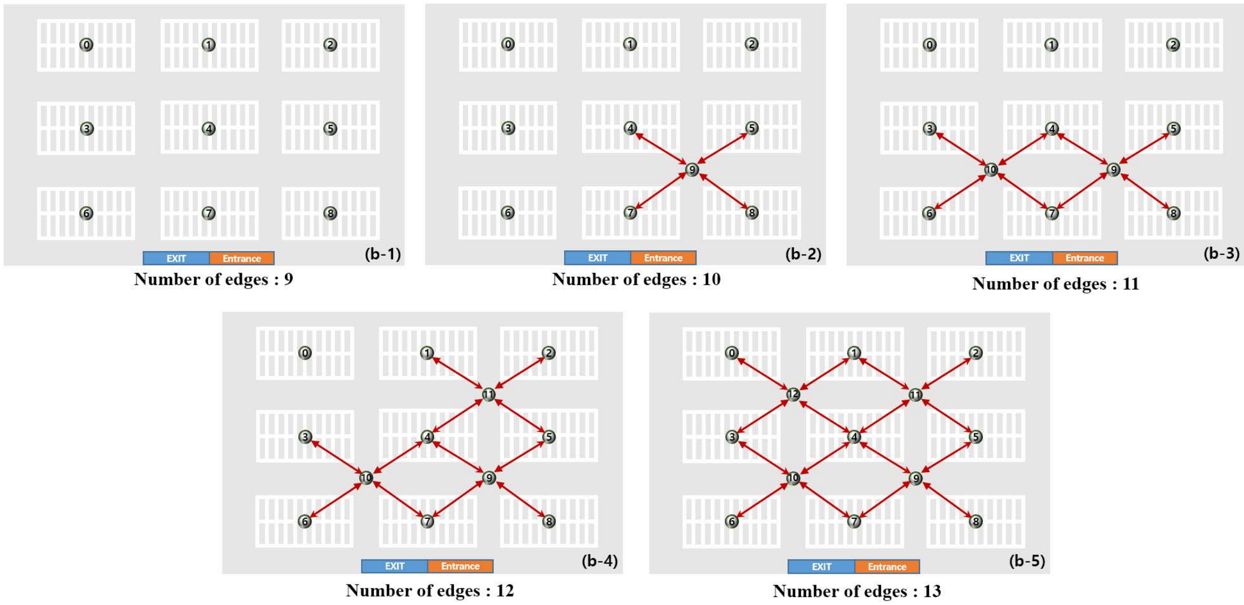

FIGURE 12. Connection of edge terminals.

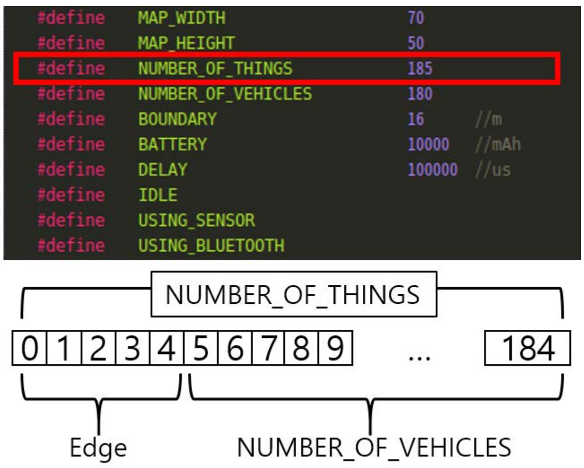

FIGURE 13. RaSim parameter settings.

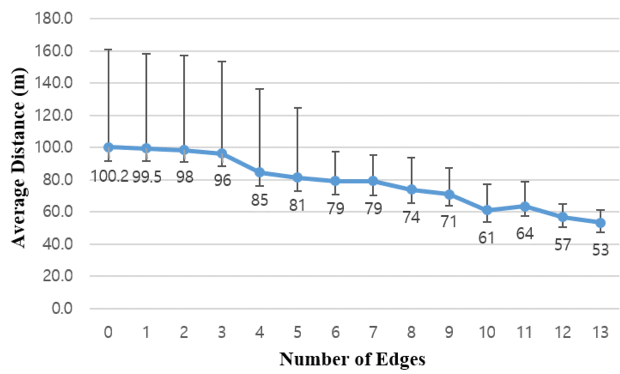

FIGURE 14. Average moving distance according to number of edge terminals.

be used to analyze communication by positions or roles of the edges and utilize basic data for communication optimization in a smart parking lot.

3) BATTERY USAGE OF EDGES

This experiment analyzed battery usage according to the number of edges. To ensure longevity, an edge should use battery power efficiently. Furthermore, even edges based on fixed power should be analyzed to reduce the maintenance cost of smart parking lots.

B. EXPERIMENT RESULT

1) AVERAGE MOVING DISTANCE AND TIME

Figures 14 and 15 show the average moving distance and time according to the number of edge terminals, respectively. Also, each figure shows the standard deviation. Through this, it can be confirmed that the deviation decreases as the number of edge terminals increases. Without edges, the smart parking lot has the same environment as a general parking lot. Thus, the highest average moving distance and time are 100.2 m and 36.07 s, respectively, in the smart parking lot without edges. With the number of edges changes from three to four, the moving distance and time are significantly reduced. The positions of the placed edges are 0, 2, 6, and 8, as shown in Figure 11(b). Thus, the mobile edge becomes capable of connecting to the edge in any direction. Because exploring empty space is easier, the width of the decrease in moving distance and time increases. With 10 edges, the distance and time decrease further because the edge can effectively explore empty spaces through the role of the spanner, as in Edge 9 in Figure 12(b-2). For 13 edges, the average moving distance is 53.43 m, and the time is 19.23 s. These values were the lowest.

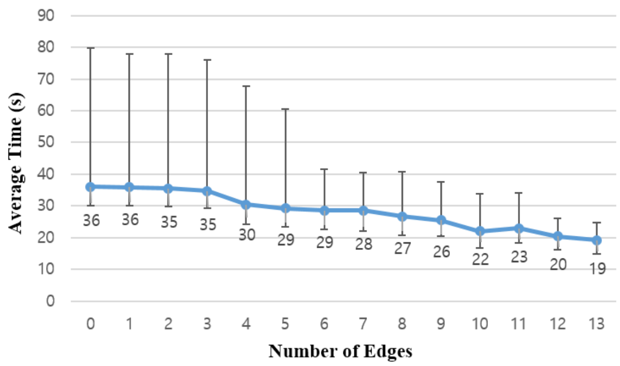

FIGURE 15. Average moving time according to number of edge terminals.

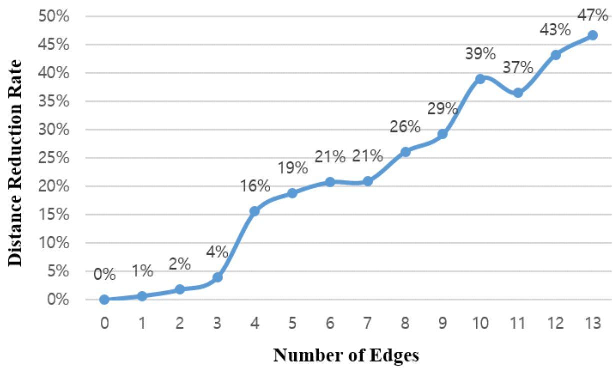

FIGURE 16. Moving distance reduction according to number of edge terminals.

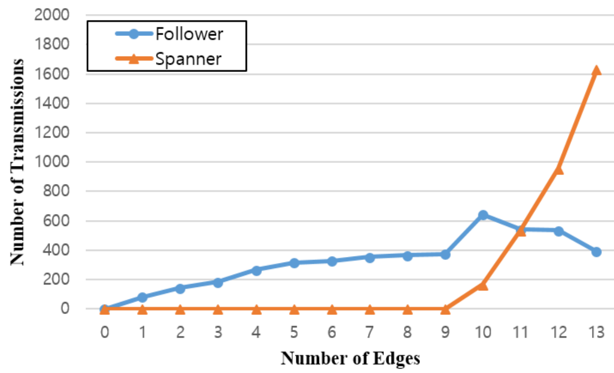

FIGURE 17. Number of data transmissions according to number of edge terminals.

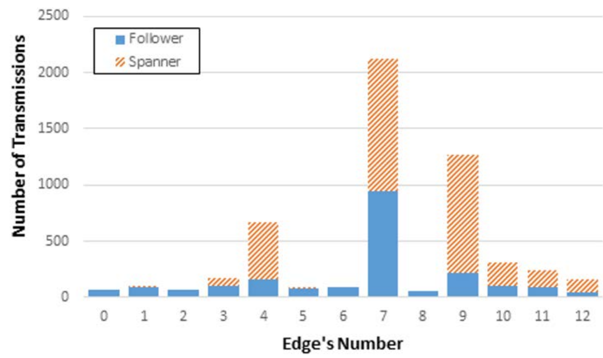

FIGURE 18. Number of data transmissions by roles.

Figure 16 shows the distance reduction rate according to the number of edges. The reduction rate was less than the moving distance with no edges, and significantly increased at four edges and ten edges. As also shown in Figure 15, the reduction rate of the moving distance increases when the moving distance decreases. Thus, the reduction rate was the highest at 47% at 13 edges.

A decrease in moving distance implies that a large amount of energy can be conserved. Therefore, smart parking lots can save energy and reduce air pollution.

2) NUMBER OF DATA TRANSMISSIONS OF EDGE TERMINALS

Figure 17 shows the number of data transmissions. Due to lack of connectivity, the spanner is not present until the number of edges reaches 10, and the number of spanner transmissions consequently remains 0. As the number of edges increases, the number of transmissions also increases and exploration becomes easier; thus, the travel distance of mobile edges decreases. There is a trade-off between battery usage and travel distance and time. More specifically, when the number of edges exceeded 10, the travel distance and time of the mobile edge significantly decreased, while the number of edge transmissions significantly increased. The spanner, which appeared when the number of edges reached 10, distributed requests to multiple locations and merged responses, which significantly increased the number of data transmissions. In figure 11(b), since the edges change to the spanner except for edge 0, 2, 6 and 8, the number of the transmissions and the battery usage by data transmissions are increased.

3) NUMBER OF DATA TRANSMISSIONS BY ROLES OF THE EDGE TERMINALS

Figure 18 shows the total number of transmissions by edges according to role. Here, Edges 0, 2, 6, and 8 do not perform the spanner role, as they only connect to a single edge, as shown in Figure 11(b). The edge that sent the most data was 7. Because Edge 7 is closest to the parking lot entrance, it is the first to connect with the mobile edge, and also performs the spanner role, which significantly increases the number of transmissions. The edge with the second highest number of transmissions was 9, due to its connectivity with Edge 7.

Edges with a large number of transmissions consume a large number of batteries. However, most of these edges can also be used as spanners. Furthermore, these edges send more data as spanners than as followers. Because spanners can significantly affect the overall network longevity, it is necessary to develop an efficient method to consume energy.

4) BATTERY USAGE OF EDGE TERMINALS

Figure 19 shows the battery usage per number of edges. We note that when the number of edges reaches 10, the battery usage sharply increases, due to the spanner increasing the number of transmissions. By analyzing only these results,

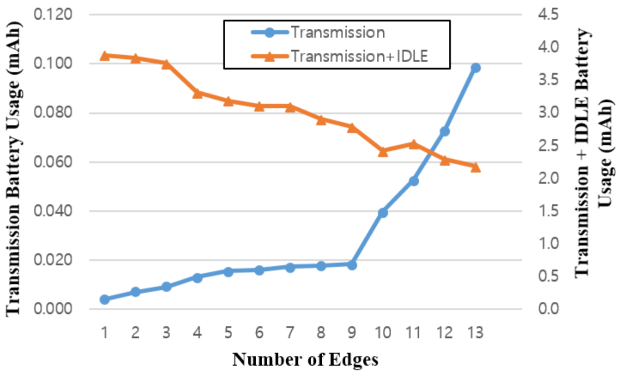

FIGURE 19. **Battery Usage according to number of edge terminals.**

it can be concluded that the presence or absence of a spanner has a significant impact on battery usage. However, all edges consume battery power even when idle, and this consumption increases over time. When accounting for idle battery usage, the total energy consumption gradually decreases. Because requests can be rapidly resolved through the spanner, the most effective number of edges was 13.

VI. CONCLUSION

In this study, we proposed a smart parking lot based on edge cluster computing for full self-driving vehicles. The proposed parking lot consists of fixed edges using Raspberry Pi as a physical model, and a mobile edge as a full self-driving vehicle.

To evaluate the smart parking lot's performance, we compared the moving distance and time in a smart parking lot with those in a general parking lot using the RaSim simulator. We also checked battery usage for data transmissions between edges and determined the most effective number of edges for a smart parking lot service. As a result of the experiment, When 13 edge terminals were placed, the smart parking lot reduced moving distance and time by 47% compared to a general parking lot. Therefore, the most effective number of edges for the smart parking lot was 13.

REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[2] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, and J. Lin, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
[3] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1697–1716, Aug. 2019.
[4] H. Peng, Q. Ye, and X. Shen, "Spectrum management for multi-access edge computing in autonomous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 3001–3012, Jul. 2020.
[5] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.
[6] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan. 2018.
[7] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–19, Feb. 2019.
[8] K. Hassoune, W. Dachry, F. Moutaouakkil, and H. Medromi, "Smart parking systems: A survey," in *Proc. 11th Int. Conf. Intell. Syst., Theories Appl. (SITA)*, Oct. 2016, pp. 1–6.
[9] T. Lin, H. Rivano, and F. Le Mouël, "A survey of smart parking solutions," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017.
[10] W. Kim and I. Jung, "Simulator for interactive and effective organization of things in edge cluster computing," *Sensors*, vol. 21, no. 8, p. 2616, Apr. 2021.
[11] W. Meulemans, M. Sondag, and B. Speckmann, "A simple pipeline for coherent grid maps," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1236–1246, Feb. 2021.
[12] J. Yang, J. Portilla, and T. Riesgo, "Smart parking service based on wireless sensor networks," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2012, pp. 6029–6034.
[13] M. Patil and V. N. Bhonge, "Wireless sensor network and RFID for smart parking system," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 4, pp. 188–192, 2013.
[14] A. Sahfutri, N. Husni, M. Nawawi, I. Lutfi, A. Silvia, and E. Prihatini, "Smart parking using wireless sensor network system," in *Proc. Int. Conf. Elect. Eng. Comput. Sci. (ICECOS)*, Oct. 2018, pp. 117–122.
[15] L. F. Luque-Vega, D. A. Michel-Torres, E. Lopez-Neri, M. A. Carlos-Mancilla, and L. E. González-Jiménez, "IoT smart parking system based on the visual-aided smart vehicle presence sensor: SPIN-V," *Sensors*, vol. 20, no. 5, p. 1476, Mar. 2020.
[16] R. Lookmuang, K. Nambut, and S. Usanavasin, "Smart parking using IoT technology," in *Proc. 5th Int. Conf. Bus. Ind. Res. (ICBIR)*, May 2018, pp. 1–6.
[17] R. K. Kodali, K. Y. Borra, S. G. N. Sharan, and H. J. Domma, "An IoT based smart parking system using LoRa," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2018, pp. 151–154.
[18] D. N. Singh and U. Divya, "An intelligent parking guidance and information system with ARM9 microcontroller," *Int. J. Latest Trends Eng. Technol.*, vol. 2, pp. 234–239, Sep. 2013.
[19] N. Bibi, M. N. Majid, H. Dawood, and P. Guo, "Automatic parking space detection system," in *Proc. 2nd Int. Conf. Multimedia Image Process. (ICMIP)*, Mar. 2017, pp. 11–15.
[20] I. M. Hakim, D. Christover, and A. M. Jaya Marindra, "Implementation of an image processing based smart parking system using Haar-cascade method," in *Proc. IEEE 9th Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Apr. 2019, pp. 222–227.

WOOJAE KIM received the B.S. and M.S. degrees from Kangwon National University, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree. His research interests include the Internet of Things, edge computing, full self-driving, and on-device AI.

INBUM JUNG received the B.S. degree from Korea University, in 1985, and the M.S. and Ph.D. degrees in computer science from KAIST, in 1994 and 2000, respectively. From 1984 to 1995, he was with Samsung Electronics Company Ltd., South Korea. He is currently a Faculty Member with Kangwon National University. His research interests include the Internet of Things, edge computing, and on-device AI.