This article has been accepted for inclusion in a future issue of this magazine.

IoT Edge-Cloud: An Internet-of-Things Edge-Empowered Cloud System for Device Management in Smart Spaces

Yoseop (Joseph) Ahn, *Student Member, IEEE,* Minje Kim, *Student Member, IEEE,* Jeongah Lee, *Student Member, IEEE,* Yiwen Shen, *Member, IEEE,* and Jaehoon (Paul) Jeong, *Member, IEEE*

***Abstract*—This paper proposes an Internet-of-Things (IoT) Edge-Empowered Cloud System (called IoT Edge-Cloud) for the visual control of IoT devices in a user's smartphone. This system uses the combination of existing technologies (e.g., DNSNA, SALA, SmartPDR, and PF-IPS), for DNS naming and indoor localization to support the visual control of IoT devices. For the visual control of IoT devices, the IoT devices register their auto-generated DNS names and the corresponding IPv6 addresses with the IoT Edge-Cloud. Each DNS name embeds an IoT device's type (e.g., fire sensor, television, refrigerator, or air conditioner) and its location information, which is obtained through an Indoor Positioning System (IPS). With the DNS name, a user's smartphone can display each IoT device and its location in an indoor place (e.g., home, office, and classroom), so that the IoT device can be located in the smartphone's screen. Through performance evaluation, this paper proposes a localization scheme for a smartphone with average localization error of 1.08 meters. Also, it proposes a localization scheme for IoT devices (especially, at the center area in a testbed) with average localization error of 1.11 meters.**

I. Introduction

THE number of IoT devices has increased from 4.8 billion in 2015 to 25 billion in 2020, showing a rapid expansion. However, about 99.4% of connectable objects in the world are currently not connected to the Internet, and the growth potential of the IoT market is very high [1]. Along with the growth of IoT devices in the market, various IoT communication and management technologies [1], [2], combined with location-based services (LBS), are entering people's lives and smartly changing the living space itself. Due to the continuous research and development on the IoT technologies, people are more willing to use IoT devices in their lives and working spaces. As a result, the number of IoT devices that an individual needs to manage is also exploding. To meet the increasing demands from IoT markets, cloud service providers such as Amazon, Google, Microsoft, and Samsung have introduced different IoT cloud services for other IoT business partners. In the developed IoT cloud services, however, less companies considered integrating visualized LBS into their services, which may greatly increase the usability of those services.

Y. Ahn, M. Kim, J. Lee, and J. Jeong are with the Department of Computer Science and Engineering, Sungkyunkwan University, Republic of Korea. (e-mail: {ahnjs124, yunminjin2, jeongahs2, pauljeong}@skku.edu). Y. Shen is with the School of Global Studies, Kyungsung University, Republic of Korea. (e-mail: chrisshen@ks.ac.kr). Note that J. Jeong is the corresponding author.

The number of IoT devices is quickly increasing, so the manual configuration of those IoT devices for management such as monitoring and remote-control is infeasible for the convenience of IoT users. The Domain Name System (DNS) can provide IoT users with the DNS names of the IoT devices for the management of those devices. To efficiently manage the information generated from those IoT devices, the information can be put into their DNS names and registered with a DNS server. Thus, a unified DNS naming for the IoT devices is important since many kinds of IoT devices are produced by the multiple vendors. That is, their DNS names need to be generated by a standardized way. To tackle the problem, DNS Name Autoconfiguration (DNSNA) [3] scheme was proposed to automatically configure DNS names for IoT devices based on certain rules. In this paper, we introduce an integrated system based on DNSNA for the proposed IoT Edge-Cloud system to efficiently manage IoT devices.

The easy use of IoT devices is another pivotal aspect for users using them in smart spaces. A smart space in this paper is defined as a space with different kinds of smart connected IoT devices, where people can live comfortably or work efficiently. A smart space can be a smart home, a smart building, a smart campus, or a smart factory. An LBS in IoT devices may provide users with much convenience for the users to manipulate those devices visually. To provision the LBS in smart spaces, it is necessary to have the location information of both IoT devices and users. This location information can be visualized in a way that the interactions become more efficient and intuitive. Different kinds of indoor localization schemes [4]–[6] can provide LBS for both IoT devices and users. However, those localization approaches have different merits and shortcomings. A pedestrian dead reckoning (PDR)-based localization approach [4] suffers from accumulated localization errors, whereas a particle filter (PF)-based approach [6] has a slow response to the movement of an object. Thus, this paper proposes an integrated localization scheme to improve the localization accuracy by combining the PDR and PF approaches.

When considering the IoT-related tasks together with the issues we have discussed, an IoT management system needs to be capable of continuously monitoring IoT devices and efficiently managing them. In this study, we introduce an IoT Edge-Cloud system based on key functions such as IoT DNS naming, indoor localization, and remote control. The proposed IoT Edge-Cloud system enables users to visually access and manage IoT devices with an indoor layout. The proposed IoT Edge-Cloud system is also intended to allow IoT devices to be controlled in a user-friendly and energy efficient manner. We use a testbed including several embedded systems for proof-of-concept (POC) of the proposed IoT Edge-Cloud system. This IoT Edge-Cloud system can process the data from both IoT users and IoT devices faster than the centralized cloud system and reduce the latency. Also, because of the distributed data processing and storage, it can improve data security and increase resilience.

II. Background

This section presents the related technologies for the proposed IoT Edge-Cloud system. The goal of our IoT Edge-Cloud system is to visually show the locations of a user and IoT devices in real time. The combination of key technologies such as IPS, DNS naming, and IPv6 address autoconfiguration can achieve this goal and the users can see their location on their own smartphone screen in real time. The key technologies for the IoT Edge-Cloud include IoT DNS naming service, PDR, indoor localization schemes, and remote control.

A. DNS Name Autoconfiguration (DNSNA) for IoT Devices

To allow users to easily manage a large number of IoT devices, IoT devices store their DNS names and IPv6 addresses in IoT DNS servers using a DNS Name Autoconfiguration (DNSNA) for IoT devices [3]. This DNSNA enables users to conveniently access IoT devices through the IoT Edge-Cloud. The overall operation process of DNSNA in the IoT Edge-Cloud is as follows. First, a DNSNA-enabled router multicasts a router advertisement (RA) with the DNS option having DNS search list (DNSSL) to an IoT device. Second, the IoT device automatically generates its DNS name according to the DNSSL information sent by the router. Third, the uniqueness of the DNS name and IPv6 address are checked by using Duplicate Name Detection (DND) and Duplicate Address Detection (DAD) mechanisms, respectively. Fourth, the router multicasts a Node Information (NI) Query message to all IoT devices to collect their DNS name and IPv6 address. Fifth, upon receiving the NI Query message, the device unicasts an NI Reply message containing its DNS name and IPv6 address to the router. When the router receives this NI reply message, it registers the DNS names and IPv6 addresses of the IoT devices with a DNS server.

Thus, DNSNA is in charge of generating the DNS name and IPv6 address for an IoT device and registering the DNS naming information of the IoT device with a DNS server. In this paper, we applied this DNSNA technique to save the information of all IoT devices and a user's smartphone into the DNS server. Each DNSNA data entry in the DNS server shows the location and unique information of either an IoT device or a user's smartphone. So, based on the information, we can track the location of both the devices and the user with a smartphone.

B. Smart Pedestrian Dead Reckoning (SmartPDR)

SmartPDR [4] is one of the key technologies in our proposed IoT Edge-Cloud. SmartPDR is an indoor localization scheme that uses inertial sensors such as an accelerometer, a gyroscope, and a magnetometer which are built into a smartphone without requiring any other expensive equipment. The smartPDR approach is working as follows. First, the walking step event detection is accomplished using the relative acceleration of the triaxial accelerometer of the smartphone. The operation for detecting a step event, which is based on the recognition of the signal pattern of the periodic inertial force, is performed by measuring the inertial force of a user's walk. Basically, a step event is detected when there is a peak in the accelerometer signal for an axis perpendicular to the ground. Second, the heading direction estimation uses the pitch, row, and yaw of the Global Coordinate System (GCS) to obtain the angle value viewed by the smartphone. Finally, for the step length estimation, the gait length is estimated based on the acceleration value of the walking. Through this process, we can estimate the location, the direction, and the speed of a user.

C. Particle Filter-based Indoor Positioning System (PF-IPS)

PF-IPS stands for Particle Filter-based Indoor Positioning System [6]. When a smartphone broadcasts Bluetooth Low Energy (BLE) beacon messages with a Universally Unique Identifier (UUID), each Anchor Point (AP) measures the Received Signal Strength Indicator (RSSI) value of each beacon message and forwards the RSSI value to an IPS server. The server uses the received RSSI values to estimate the location of the smartphone using an algorithm called Particle Filter (denoted as PF). The PF algorithm creates virtual particles to represent where the smartphone may be located and calculates the weights of the particles with the RSSI information of BLE beacons received by APs. Since the RSSI information received by APs is very unstable, it is refined through a Kalman Filter (KF) to be more stable [6]. With the particles and their weights based on RSSI values, the location of the smartphone can be obtained by averaging the coordinates of particles.

SmartPDR and PF-IPS techniques can be combined to effectively track a user's location in the IoT Edge-Cloud. The error value accumulated in the SmartPDR can be periodically corrected by the position from PF-IPS as a landmark, leading to a high localization accuracy. PF-IPS can also use the heading information of a smartphone from SmartPDR to improve the movement accuracy of particles, which can improve the positioning accuracy and reduce its convergence time. Therefore, this fusion approach benefits from the fast location computing by SmartPDR and the improved location estimation of the user by the PF-IPS with the help of the estimated heading direction information provided by SmartPDR.

D. Smartphone-Assisted Localization Algorithm (SALA)

SALA is a technology designed to estimate the location of an IoT device [5]. SALA measures the location of the IoT device based on the location of the smartphone (using

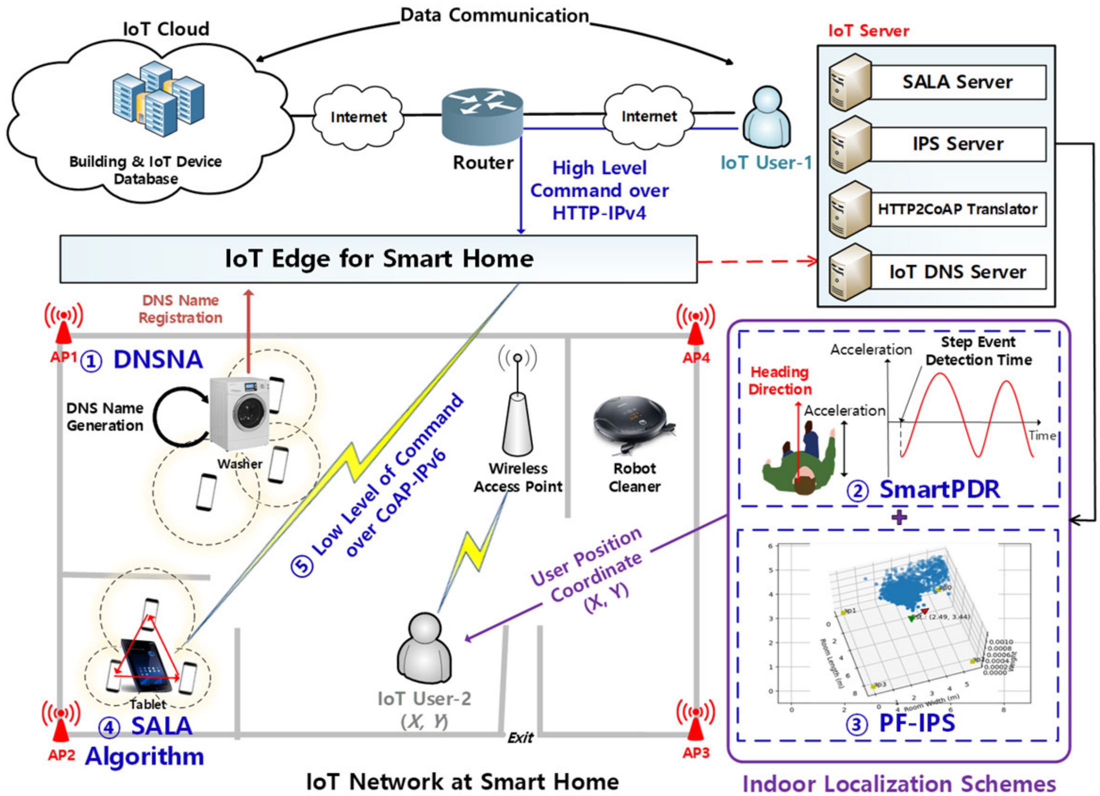

Fig. 1. The workflow of the IoT Edge-Cloud system.

PDR) as well as the RSSI values of the messages exchanged between a smartphone and an IoT device. While a user walks around an IoT device with a smartphone, the smartphone periodically broadcasts either a WiFi or BLE beacon message containing its location information [7]. When receiving this beacon message, IoT devices send a SALA server their own device information, the smartphone's location, and the RSSI value for such a beacon message that is received by each IoT device. Thus, SALA algorithm can automatically update the location information of IoT devices with the data received from the IoT devices.

E. IoT Control Message Protocols

Constraint Application Protocol (CoAP) [8] is a specialized web transfer protocol for constrained nodes (e.g., IoT devices) and constrained (e.g., low-power and lossy) networks. CoAP is a RESTful protocol that has the architectural style of Representational State Transfer (REST) like HTTP. It works above a simple transport layer like User Datagram Protocol (UDP) to control IoT devices such as low-power sensors, switches, and valves. In an IoT Edge-Cloud, CoAP will serve as a communication protocol that delivers IoT control commands from users to IoT devices. When a user selects a list of commands with which to control an IoT device using a smartphone, the smartphone transmits a control message to the IP address of the IoT device. Message Queuing Telemetry Transport (MQTT) [9] can be used as another type of message protocol based on a publish-subscribe architecture for Machine-to-Machine (M2M) and IoT device communications. MQTT can facilitate communications between IoT devices and users with less power and fewer packets by the mechanism of subscribing and publishing. In the middle of the mechanism, a broker as a message publisher distributes messages to subscribed clients, thus allowing a large number of sensor devices to efficiently report information to the clients without duplication. However, while CoAP operates on UDP, MQTT operates on Transmission Control Protocol (TCP), which requires a long-term connection that consumes more power and network resources. Due to this reason, using MQTT can shorten the battery life of an IoT device. Further, since messages can only be managed by a broker, the message forwarding by the broker can be interrupted if the broker faces problems such as power outage and network attacks. Therefore, our proposed IoT Edge-Cloud uses CoAP instead of MQTT, with the consideration of a lightweight network while being able to manage a large number of IoT devices.

III. Design of IoT Edge-Cloud

This section presents the architectural design of the IoT Edge-Cloud. With the emergence of the IoT era, it has become increasingly difficult to manually manage the growing number of IoT devices from various vendors. Therefore, we propose a system that can control and manage various IoT devices with standardized protocols in a unified way within a cloud-based system. The goal of this paper is to visually show the locations of a user and IoT devices in real time. As shown in Fig. 1, the proposed IoT Edge-Cloud system for a smart home can mainly consist of a user's smart device (e.g., smartphone and tablet), an IoT Cloud, an IoT Edge System, and IoT Devices. The IoT Edge System may host several IoT servers such as SALA server, IPS server, and IoT DNS server by shared computation resources.

• **A User's Smart Device**

A user's smart device (e.g., smartphone) as the IoT User-1 or User-2 shown in Fig. 1 basically needs to know the global IP address of the IoT Edge-Cloud Server. The smartphone of a user can show a building's information when receiving information from the IoT Edge-Cloud for a building structure map and IoT device lists. The smartphone uses the BLE technology to continuously advertise BLE beacon messages for performing operations of PF-IPS and SALA techniques. In this way, we can show the locations of IoT devices on the screen of our smartphone according to SALA data. The SmartPDR and PF-IPS techniques are also used to continuously track the location of the smartphone and show it on the screen of the smartphone. The user may graphically control a desired IoT device by sending a control message toward the router to which the IoT device belongs.

• **IoT Cloud**

An IoT Cloud entity stores the information of buildings and IoT devices used by IoT users, as shown in Fig. 1. The smartphone of the user can obtain the information about buildings, floors, and rooms from the IoT Cloud when the user accesses his or her room. The IoT Cloud also sends the IP address of the IoT Edge System to the IoT devices in each room; therefore, user smartphones can have the information about the IoT devices by accessing the IoT servers in the IoT Edge System.

• **IoT Edge**

The IoT Edge manages all the technical processes (i.e., DNSNA for DNS naming, SALA for device localization, IPS for user localization, and CoAP translation service) as edge computing servers (also called IoT servers) in a smart space. For tracking the location of a user having a smartphone, APs in the smart space send the received BLE beacon information to the IoT Edge for the IPS function. The PF-IPS Server inside the IoT Edge thus can calculate and track the locations of the user. When the user needs to control an IoT device, a control message can be converted into a CoAP message upon arriving at the HTTP2CoAP Translator inside the IoT Edge, and then delivered to the target IoT device.

• **IoT Device**

IoT devices are all kinds of devices that improve work efficiency and comforts in smart spaces. The IoT devices must be connected to the internal network of a router through IPv6. Since CoAP is used, low-power IoT devices (e.g., temperature sensors and fire detectors) may also be connected to the network. An IoT device can accept a control message sent by the router and then perform a command in the control message. If the command is successfully performed, the IoT device sends the user smartphone a success status as a response.

Fig. 2 shows a sequence diagram for the proposed IoT Edge-Cloud system. Before a user can send control messages to an IoT device, several steps shall be done as follows. First, a user's smartphone App can access the IoT Cloud through HTTP. The App can automatically fetch the current building information from the IoT Cloud based on the estimated outdoor location of the user. The estimated location in the App can be determined by an infrastructure-based location service such as cellular networks or wireless LAN. The user can also manually select a target area (e.g., a campus) and the current building. Once the building information is obtained from the IoT Cloud, the user can choose the floor and the room number of the building to access. Next, when a user enters the room with the selected room number and requests the IoT device list in the room, the request is transmitted to the IoT Edge to obtain the IoT device list, which includes the IPv6 addresses and DNS name information for all devices available in the room. Eventually, when the user receives the IoT device list, the room layout is also displayed on the smartphone's screen with icons for IoT devices at their locations according to the embedded physical location information in the DNS names of the IoT devices.

At this moment, the user is ready to control the IoT devices in the room. The user can select an IoT device icon to control. Once a control command is executed, a control message embedded into an HTTP message is sent to the HTTP2CoAP Translator [10]. The translator converts the received command to the format of a CoAP control message and then delivers it to the IoT device with its IPv6 address. Upon receiving the CoAP message, the IoT device can perform the requested command, and returns an execution result to the user's smartphone such that it can visually display the status of the IoT device.

IV. Implementation of IoT Edge-Cloud

This section describes the implementation of the proposed IoT Edge-Cloud based on an enhanced indoor localization service. We set up an implementation environment that has three IoT devices, a Raspberry Pi 3 Model B (Pi3B) unit as a router with a DHCP server and an RADVD (i.e., Router Advertisement (RA) Daemon) [11], six Raspberry Pi 4 model B (Pi4B) units as APs for PF-IPS, and three Pi3B units for DNSNA.

• **RADVD:** The Router Advertisement Daemon (RADVD) is an open-source software that generates link-local advertisements with IPv6 router addresses and routing prefixes by the Neighbor Discovery Protocol (ND). When IPv6 hosts configure their network interface cards (NIC), they multicast router solicitation (RS) requests onto the network to discover available routers.

• **DNSNA:** We implemented DNSNA in Raspbian operating system of Pi3B. When an RA message is sent by RADVD, an IoT device receives it and then generates a DNS name based on the DNSSL along with its physical location. The generated DNS name is verified through the DAD process, and the router registers the DNS name information of the IoT device with a DNS Server after verification.

• **IoT Cloud:** We implemented the IoT Cloud based on a desktop PC. The IoT Cloud stores location information such as sections, buildings, floors, and room numbers. An App in the smartphone of a user can receive a building structure map from the IoT Cloud via HTTP.

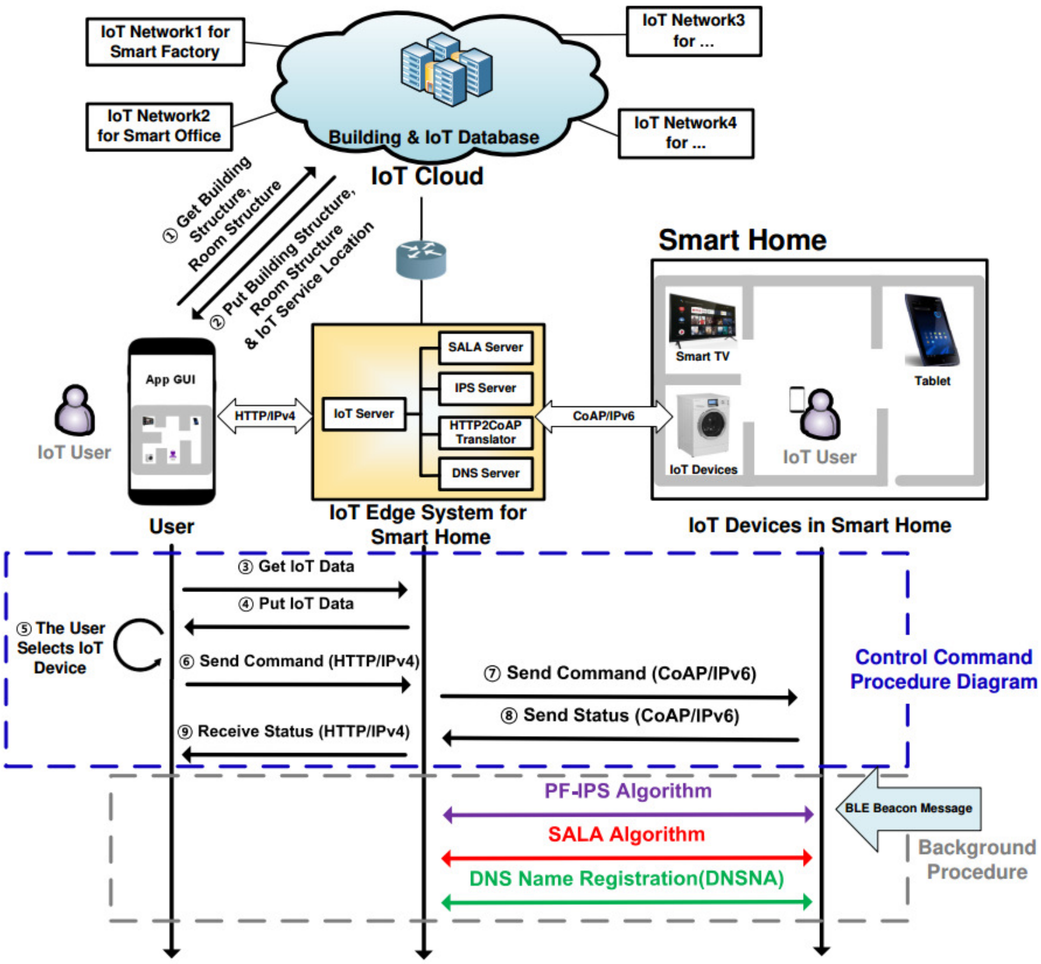

Fig. 2. The sequence diagram of the IoT Edge-Cloud system.

- **Smartphone Application:** We implemented an Android smartphone App that imports HTTP communication modules to communicate with the IoT Edge-Cloud and gateways, along with Bluetooth modules imported for various sensor listeners and BLE.
- **Anchor Point (AP):** We used Pi4B units for all APs, and the RSSI parameters were determined through the RSSI calibration process. Each AP runs a BLE collection process and communicates with the PF-IPS server.
- **PF-IPS:** We implemented the PF-IPS in Raspbian of a Pi4B. When opening a socket and receiving BLE beacon messages from a user's smartphone, an AP sends data having beacon messages and RSSI values to the PF-IPS server. The server accumulates the data, calculates the location of the user's smartphone based on the data, and transmits the location estimate to the smartphone.
- **SALA Algorithm:** To improve the performance of our SALA algorithm [5], we implemented the new data communication part and location calculation part of the SALA algorithm, respectively. When the SALA server starts running, it uses all the reported data and updates the position of an IoT device. While a user walks around an IoT device with a smartphone, the smartphone periodically broadcasts a BLE beacon message containing its location information. When receiving this beacon message, IoT devices send a SALA server their own device information, the smartphone's location and the RSSI value for such a beacon message that is received by each IoT device. Thus, SALA algorithm can automatically update the location information of IoT devices with the data received from the IoT devices.
- **CoAP and HTTP2CoAP Translator:** We implemented the CoAP communication part and HTTP2CoAP Translator based on a CoAP open source framework called Eclipse Californium [12]. The purpose of using HTTP2CoAP is to convert HTTP-based messages (i.e., HTTP request and response) into CoAP-based messages (i.e., CoAP request and response). The performance gain is that IoT devices can save much energy by not processing HTTP-based messages. In most cases, they cannot process HTTP-based messages due to their limited resources.

V. Performance Evaluation

This section describes a performance evaluation that was conducted to evaluate two kinds of IPS schemes in the proposed IoT Edge-Cloud. The first IPS scheme is for the localization of a user's smartphone. It can be SmartPDR [13], PF-IPS [6], or the combination of SmartPDR and PF-IPS. The second IPS scheme (called SALA [5]) is for the localization

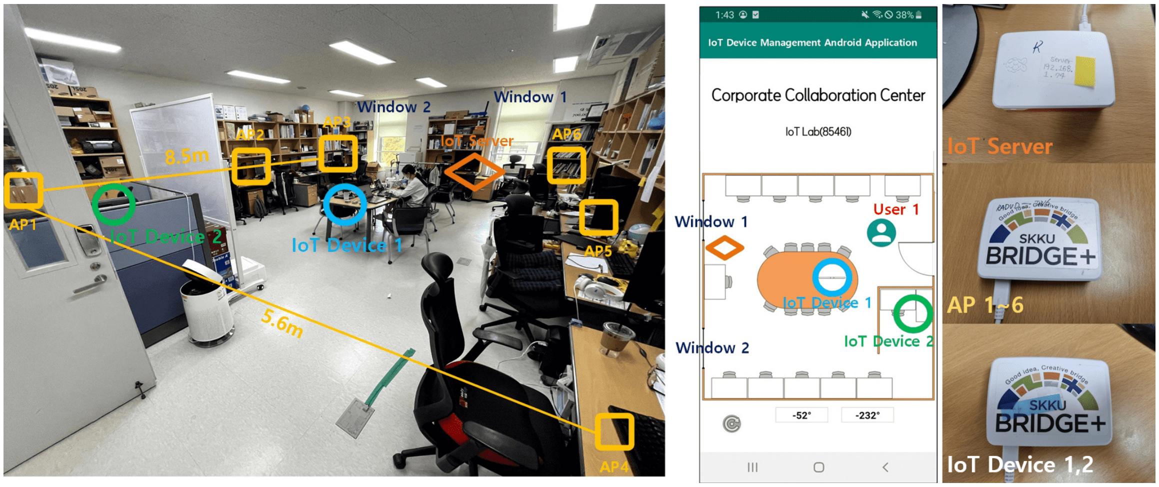

Fig. 3. Testbed and smartphone App for IoT device localization.

of IoT devices. The main goal of this paper is to visually show the locations of the user's smartphone and IoT devices in real time by reducing the localization errors of them. Thus, the integration of SmartPDR, PF-IPS, and SALA can support this goal effectively.

- **Testbed**

We experimented our proposed IoT Edge-Cloud system in a space with 8.5 m $\times$ 5.6 m. The experimental environment is similar to a normal working space having tables, chairs, and other facilities in the environment. The user smartphone used for the experiment is LG LM-V500N model with Android 10 installed. The version of Bluetooth in the smartphone is 4.0, and the Tx power and the TX frequency (i.e., packet interval) of Bluetooth are set to -56 dbm and 100 ms, respectively.

Fig. 3 shows the testbed with an IoT server for an IoT Edge-Cloud system, APs, and IoT devices along with a smartphone App. Six APs were installed along the walls. In a larger space, more APs can be used to improve localization accuracy. A picture of a screen shot of IoT devices is added to Fig. 3 for their depiction. A Pi3B unit as an IoT device was placed on the desk, and the other two Pi3B units were used as the IoT server and the RADVD, respectively. After initiating on a smartphone, the dedicated App communicates with the IoT Edge-Cloud system to receive the layout of the room. After the location of the IoT device is estimated, an indoor layout of the room with the location of the IoT device is displayed on the App, as shown in Fig. 3. The location of the smartphone can be calculated and updated through the combination of both SmartPDR and PF-IPS.

When selecting the icon of the IoT Device, the user can control the IoT Device by sending an HTTP command toward the router. The router translates the HTTP message into a CoAP message as an HTTP2CoAP Translator and sends the CoAP message to the IoT Device. The IoT Device then executes the command and sends its response to the smartphone. Note that the source code and documents of the IoT Edge-Cloud are available at https://github.com/jaehoonpauljeong/IoT-Edge-Cloud, and a video demonstration is available at https://www.youtube.com/watch?v=ZzA7dDJLuHk.

- **Performance Metric**

We use the localization error as a performance metric by measuring the Euclidean distance between the actual and the estimated positions of the smartphone. After repeating the experiment 10 times, the mean and standard deviation of each point in the x-axis are computed for an error bar indicating a 95% confidence interval.

We measured the RSSI values of some points in the room to determine the parameters of the designed Kalman filter. Starting from the center of the room, increasing the distance by 50 cm in four directions, east, west, south, and north, we measured the RSSI values of sample points from the center of the room until we reached the wall. The parameters of the designed Kalman filter are determined by the actual measurement of the experiment environment, which is explained in detail by the work in [6]. For different indoor environments, the parameters need to be updated through a calibration process. The accuracy of a smartphone's location is calculated based on the several smartphone sensors such as accelerator, gyroscope, and magnetometer. The smartphone's location is calculated based on the Kalman filter-calibrated graph. We measured RSSI values in the room, and calculated the RSSI value graph using the Kalman filter. As shown in the first figure of Fig. 4, the locations of the smartphone by the three localization schemes have some location errors. This figure shows how much localization error occurs on every step for each localization scheme. The localization error of the IoT devices can be found in the second figure of Fig. 4. There is some localization error range on the figure, and this range shows the variation of the localization error.

- **Performance Analysis**

As shown in the first figure of Fig. 4, the x-axis shows the number of data collections, and the y-axis shows the localization error (named Smartphone's Localization Error) of the smartphone. The data is collected once every 2.5 seconds for a total of 125 seconds. The first figure of Fig. 4 shows a smartphone's localization errors with error bars of PF-IPS, SmartPDR, and the integrated approach (i.e., a fusion of PF-

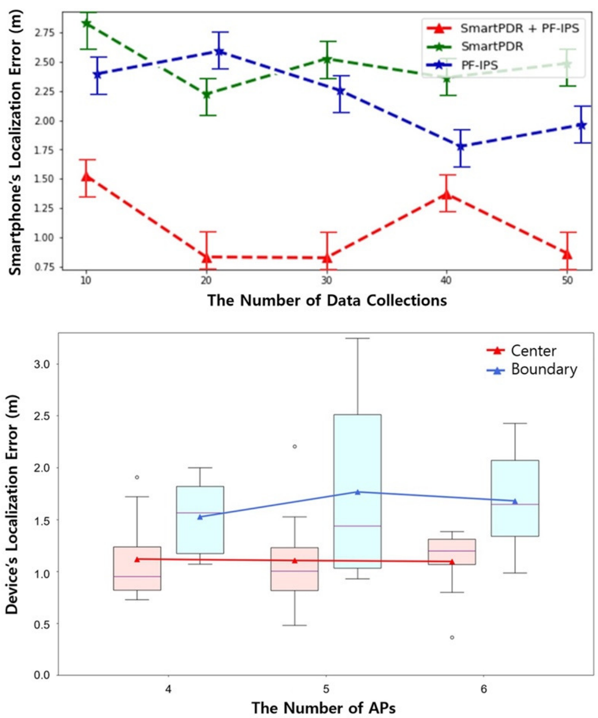

Fig. 4. Performance of the localization schemes in the IoT Edge-Cloud system.

IPS and SmartPDR). We chose x-axis as the number of data collections to show that the number of the data collections does not affect the performance of localization significantly. Even though the number of the data collections increase, the performance of each technique does not change a lot.

As shown in the first figure of Fig. 4, among the three approaches, the combination of SmartPDR and PF-IPS shows the best localization performance with an average localization error of 1.08 m and an average standard deviation of localization error of 1.01 m. On the other hand, PF-IPS has the localization performance with an average localization error of 2.25 m and an average standard deviation of localization error of 1.09 m. SmartPDR has the localization performance with an average localization error of 2.49 m and an average standard deviation of localization error of 1.34 m. Thus, according to our experiment results, the combination of SmartPDR and PF-IPS shows a reduced localization error by 51.7% over PF-IPS, and by 56.4% over SmartPDR in average.

The second figure in Fig. 4 shows the impact of different numbers of APs on the localization errors when the IoT device exist in the experiment room. In this figure, as the number of the APs increases, the average and range of localization errors of IoT device located on the boundary region (denoted as Boundary) trend to fluctuate, but the average and range of localization errors of IoT device located at the center area (denoted as Center) trend to decrease.

Thus, from Fig. 4, we can see that, when the IoT device is located at the boundary regions, the localization errors are higher than when the device is located at the center area of the experiment environment of Fig. 3.

As shown in the second figure of Fig. 4, an IoT device (i.e., IoT Device 1 in Fig. 3) located at the center has the localization performance with an average localization error of 1.11 m and an average standard deviation of localization error of 0.45 m. On the other hand, an IoT device (i.e., IoT Device 2 in Fig. 3) located at the boundary has the localization performance with an average localization error of 1.66m and an average standard deviation of localization error of 0.61 m. It is observed that the IoT device at the center has 33.1% reduced localization error than the IoT device at the boundary.

This phenomenon can be caused by two reasons: 1) strong reflections of wireless signals transmitted by the IoT device near the boundary region and 2) weak wireless signals received by the APs located at the far end of the experiment environment for localization.

For the first reason, when the IoT device is located at the boundary regions, the walls strongly reflect the wireless signals transmitted by the IoT device. The APs near the IoT device received much stronger wireless signals and more reflected interference signals from the IoT device, which caused a higher localization error.

For the second reason, since the IoT device is located at the boundary regions, the wireless signals transmitted by the IoT device become weak for those APs located at the far end of the experiment environment, which causes those APs to generate a biased distance estimation. Thus, the above two reasons cause higher localization errors for the IoT device located at the boundary regions. Because of these reasons, SALA localization error have fluctuations and the range boundary of localization error can be checked in the first and the second graph in Fig. 4.

When the user is moving around the center area of the room, the average localization error is about 1.11 m with less variances. When the user, however, is moving at the boundary area of the room, the average localization error of the smartphone is increased to about 1.66 m with larger variances. From the perspective of the increasing number of APs, though the localization error of the center nodes shows an almost constant performance with the increasing number of APs, the variation (i.e., standard deviation) of the localization error becomes smaller. It indicates that more APs may provide a more stable localization accuracy. Based on the results in our experiment, we can see that the localization performance of the proposed IoT Edge-Cloud system is robust to different factors such as the number of APs and that of IoT devices.

VI. Research Challenges

In this section, we describe the research challenges for the real-world applications of the proposed IoT Edge-Cloud system.

• **Indoor Layout Construction** To display the locations of IoT devices on a smartphone, an indoor layout is required. For easing the preparation of an indoor layout, an indoor map generation scheme needs to be provided to obtain the layout of the house, which can be done by a robot cleaner with LiDAR or the building layout database from the construction company of the house. Thus, an efficient approach to generate and correct the indoor layout can largely improve the system applicability in the real world.

• **RSSI Value Instability** SALA and PF-IPS use the RSSI values of beacon messages sent by a user's smartphone for localization. However, RSSI values measured at a receiver usually experience much noise, causing the fluctuations of the RSSI data every time it is measured. In the proposed IoT Edge-Cloud, we have designed a Kalman filter to reduce the fluctuations, which shows a good performance. Therefore, it is imperative to study and incorporate localization methods not only using RSSI but also other means such as the channel state information. We need to do more research to evaluate the environmental conditions, identify the communication protocols, select suitable devices, optimize power consumption, and ensure network connectivity. With these kinds of techniques, we can extend the IoT Edge-Empowered Cloud system to the outdoors as well as the indoors.

• **Secure Communication** To protect a user's privacy, the communication among different entities in the proposed IoT Edge-Cloud should be securely protected. Non-secure communication is vulnerable to security attacks such as eavesdropping. The communications can be protected by Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), so the HTTPS message should be translated into a CoAP message over DTLS through key management via the IoT Edge-Cloud. For these issues, designing a secure communication protocol for IoT devices becomes a pressing research topic. For IoT security, Artificial Intelligence (AI) can be one of the methods to enhance the security for the IoT devices in user authentication, access control, and data offloading [14].

• **Multiple User Management** The current proposed IoT Edge-Cloud does not implement the registration and authentication management for multiple users. Users in the same household should be grouped together, and their devices should be able to be managed collectively. Research on schemes for efficient user registration and authentication can accelerate the adoption of IoT devices. Also, privacy protection should be considered in the tracking of multiple IoT users with smartphones.

VII. Conclusion

This paper introduces an IoT Edge-Cloud that allows for the remote-control of IoT devices by an IoT user using a smartphone. The IoT Edge-Cloud supports the visualization of the IoT devices and a user's smartphone in an indoor space for the graphical control of the IoT devices by the smartphone. For these remote control and visualization services, we calculate the location of the user in the room based on the DNSNA, SmartPDR, and PF-IPS. The locations of the IoT devices are calculated by SALA. We evaluated the proposed localization schemes and showed their effectiveness. As future work, we will support secure communication of HTTP and CoAP with TLS and DTLS in the IoT Edge-Cloud System.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea MSIT (No. 2022-0-01199 and No. 2022-0-01015).

References

[1] S. Sinche, D. Raposo, N. Armando, A. Rodrigues, F. Boavida, V. Pereira, and J. S. Silva, "A Survey of IoT Management Protocols and Frameworks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1168–1190, May 2020.

[2] A. E. Braten, F. A. Kraemer, and D. Palma, "Autonomous IoT Device Management Systems: Structured Review and Generalized Cognitive Model," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4275–4290, Mar. 2021.

[3] K. Lee, S. Kim, J. P. Jeong, S. Lee, H. Kim, and J.-S. Park, "A framework for DNS naming services for Internet-of-Things devices," *Elsevier Future Generation Computer Systems*, vol. 92, pp. 617–627, Mar. 2019.

[4] M. T. Ahammed, U. M. Boppana, R. Kondapalli, M. D, P. Balaji, and M. A. Zardar, "Survey on Wireless Fidelity (Wi-Fi) Based Indoor Location Technology for Mobile Devices," in *Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, Aug. 2022, pp. 1465–1470.

[5] J. P. Jeong, S. Yeon, T. Kim, H. Lee, S. M. Kim, and S.-C. Kim, "SALA: Smartphone-Assisted Localization Algorithm for Positioning Indoor IoT Devices," *Springer Wireless Networks*, vol. 24, pp. 27–47, Jan. 2018.

[6] Y. Shen, B. Hwang, and J. P. Jeong, "Particle Filtering-Based Indoor Positioning System for Beacon Tag Tracking," *IEEE Access*, vol. 8, pp. 226 445–226 460, Dec. 2020.

[7] S. Sophia, B. M. Shankar, K. Akshya, A. C. Arunachalam, V. T. Y. Avanthika, and S. Deepak, "Bluetooth Low Energy based Indoor Positioning System using ESP32," in *Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Sep. 2021, pp. 1698–1702.

[8] M. A. Wicaksono and Y. Bandung, "CoAP Evaluation for Node-Edge Communication on Wireless Multimedia Sensor Network," in *International Conference on Information Technology Systems and Innovation (ICITSI)*, Nov. 2022, pp. 122–127.

[9] B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey," *IEEE Access*, vol. 8, pp. 201 071–201 086, Nov. 2020.

[10] P. Pierleoni, A. Belli, L. Palma, L. Incipini, S. Raggiunto, M. Mercuri, R. Concetti, and L. Sabbatini, "A Cross-Protocol Proxy for Sensor Networks Based on CoAP," in *IEEE 23rd International Symposium on Consumer Technologies (ISCT)*, Jun. 2019, pp. 251–255.

[11] J. Tang, "Research on IPv6 Protocol Transition Mechanism," in *6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, Apr. 2021, pp. 702–705.

[12] Eclipse Foundation. Californium (Cf) - CoAP for Java. (Accessed on Apr. 26, 2023). [Online]. Available: https://github.com/eclipse/californium

[13] W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2906–2916, May 2015.

[14] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions," *Mobile Networks and Applications*, pp. 1572–8153, Mar. 2022.

Yoseop (Joseph) Ahn is a Ph.D student in the Department of Computer Science and Engineering, Sungkyunkwan University, Korea.

Minje Kim is an undergraduate student in the Department of Computer Science and Engineering, Sungkyunkwan University, Korea.

Jeongah Lee is an undergraduate student in the Department of Computer Science and Engineering, Sungkyunkwan University, Korea.

Yiwen Shen is a research professor in the School of Global Studies, Kyungsung University, Korea.

Jaehoon (Paul) Jeong is an associate professor in the Department of Computer Science and Engineering, Sungkyunkwan University, Korea.