IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2019.DOI

Anti-Collision Voting Based on Bluetooth Low Energy Improvement for the Ultra-Dense Edge

DAOQI HAN[1], LINGYI XU[1], RUOHAN CAO[1], HUI GAO[1] AND YUEMING LU.[1]
[1]Beijing University of Posts and Telecommunications, Beijing, P. R. China

Corresponding author: Yueming Lu(e-mail: ymlu@bupt.edu.cn).

This work was supported by the Key Technologies of Trusted Sharing of Multi-source and Multi-modal Data Based on Blockchain Project under grant No.2019YFB2102403

ABSTRACT Due to the popularity of short-range communications and the ultra-dense deployment of the Internet of Things (IoT), the occurrence of severe signal collision in crowded nodes is becoming increasingly critical. We propose a scheme called anti-collision voting based on Bluetooth low energy improvement for the ultra-dense edge, in which one topic can be identified and separated from the others to enable intensive voting. The scheme alleviates the exponential growth of advertising collisions and achieves efficient consensus between the devices and humans by localized the message advertising. The previous proposals focused on enhancing the hardware and optimizing the parameters to establish a connection for interaction. Considering the limitation of connective devices, our scheme adopts a novel advertising model in which the nodes can promptly exit the advertising state by actively responding. In addition, the scheme merges active responses into a group to notify the advertising nodes and randomly chooses a channel to reduce redundancy and save energy. The simulations demonstrate that the scheme can reduce the collisions by more than 90% and channel occupancy by half, while doubling the capacity and efficiently supporting intensive voting within edge networks.

INDEX TERMS Anti-Collision, BLE, Edge Computing, iBeacon, IoT.

I. INTRODUCTION

WITH the development of the Internet of Things (IoT) technology, direct communication between devices (D2D) has become a research focus area [1]. D2D exhibits advantages of latency, dynamic neighbor discovery, and high spectral efficiency over short ranges. At present, the widely used Quick Response (QR) code [2] cleverly adopts image recognition, which eliminate any communication overhead; however, the code position must be manually determined. In the short-range communications field, the inherent confidentiality of sonic technology can facilitate lightweight security authentication [3]. Moreover, the Bluetooth technology [4] has achieved breakthroughs in terms of high speed, low power consumption, and rapid neighbor discovery. Drawing on the successful experience of Bluetooth, 5G technology adds high-speed D2D communication to a cellular architecture [5] to solve ultra-dense deployment and latency issues.

Building a distributed voting mechanism based on high-speed D2D communication can promote additional members to join the decision-making process. Furthermore, we need a transparent voting mechanism to reduce extremes and manipulation behaviors. Based on edge computing, this multi-party voting can efficiently achieve consensus between devices and humans. In this manner, we can implement a lightweight ledger for data protection through multiparty verification and multistep confirmation such that every entity can save its arguments and evidence. These consensus results facilitate the formation of long-term stable strategies for organizations. In addition, we are motivated by the need for reliability of data sources with edge nodes and the requirement of an authenticity mechanism involving zero-knowledge proof [6].

The iBeacon protocol exhibits several advantages such as low power consumption and long-distance coverage [7]. By using Bluetooth Low Energy (BLE) technology to broadcast packets, devices can promptly discover information points, trigger specific applications, positioning, and interactions within a 100-meter range [8]. Thus, this technology is mainly applied in open areas such as malls, libraries, museums, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

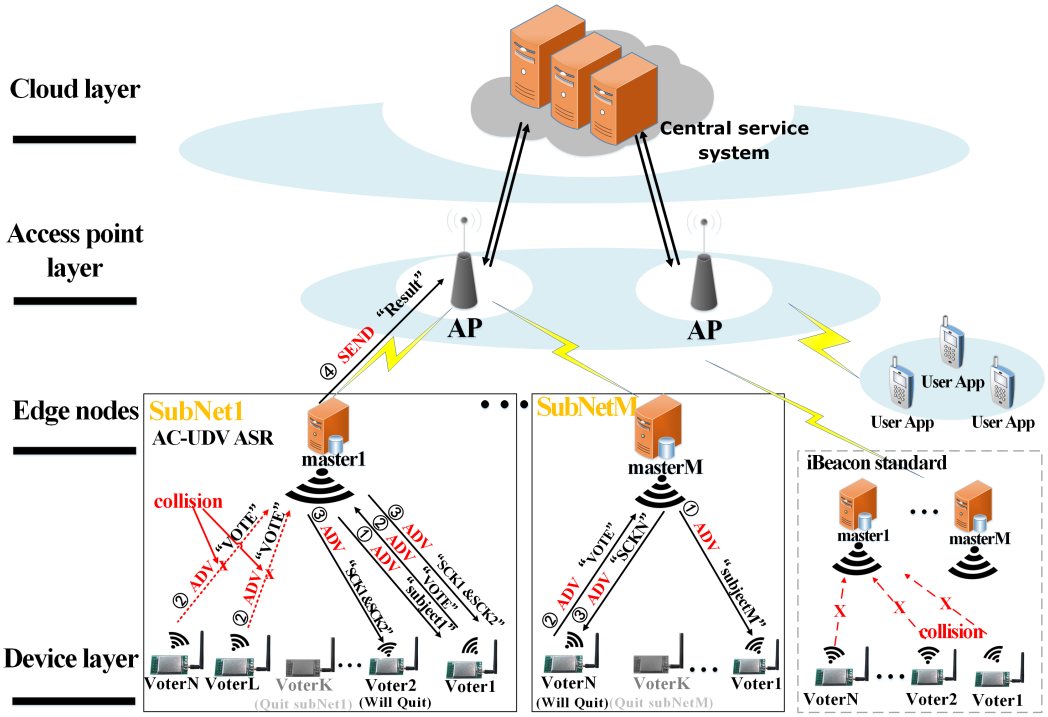


FIGURE 1. Typical iBeacon advertising protocol and voting process. The dashed box shows the traditional voting process, which establishes connections based on advertising. The solid box shows the new protocol based on the subnet isolation and the response.

conference venues [7]–[12]. Nevertheless, it is necessary to enhance the neighbor discovery efficiency and achieve low latency and rapid user responses [13]. Furthermore, recent studies highlighted the occurrence of advertising collision in a dense IoT environment, in which a large number of concurrent applications considerably deteriorate the performance [14], [15]. The anti-collision advertising protocols [16]–[21] can help reduce the latency and increase the capacity.

To alleviate the high channel occupancy problem in the BLE connectionless mode, we first review the communication process using the iBeacon protocol (see Fig. 1). The dashed line box in Fig. 1 represents a traditional BLE network that includes M master nodes and N slave nodes [22]. The master node performs listening in the network, responds to advertising from the slave nodes, and establishes connections with these nodes. The original scheme builds a reliable connection that is suitable for scenarios involving considerable data. However, when the number of nodes in the network is large or multiple masters simultaneously attempt to respond to the same slave node, channel collisions occur, leading to packet losses. In this example, the foremost issue pertains to the repeated advertising by the slave nodes. As shown in the solid boxes in Fig. 1, a novel connectionless network is suitable for the master-slave scenario based on iBeacon with a small number of merge-confirmation responses labeled as ③ ADV "SCK1 & SCK2". The main processes can be described as follows: ① The master advertises its existence. ② The slaver advertises its own data when it finds that the master exists. ③ The master scans and receives the data and simultaneously sends a voting response to multiple slavers. The slaver receives the response and exits the network in time. ④ After confirming that the topic voting is complete, the master sends the voting results to the cloud platform.

Next we investigate advertising collisions in dense environments. The existing solutions focus on dynamic intervals [21], [23]–[25] and random backoffs [26]–[28].

With reference to these studies, we establish a new protocol with the BLE parameter optimization, using an ARM microcontroller to perform message advertising with an embedded low-power Bluetooth framework based on the iBeacon protocol, thereby enabling Anti-Collision Voting for the Ultra-Dense Edge (AC-UDE). The prototype and testbed are shown in Fig. 2. The master can actively and simultaneously respond to multiple voter nodes. After the voters receive the response, they quit the network, thereby reducing the redundant broadcasts. The experimental results show that compared with the non-response advertising protocol, the proposed AC-UDE can reduce the latency by 23% and advertising collisions by 90%, thereby doubling the capacity.

Unlike the voting based entirely on the cloud platform, the new scheme introduces smart edge nodes to control the anti-collision, cache voting status, and aggregate the contents to reduce the interaction with the cloud servers. The major contributions of this research can be summarized as follows:

- Based on the BLE connectionless mode, we implement an efficient and transparent active response mechanism to enable large-scale voting with more than 200 nodes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

- Through a node-state-switching process, we establish a performance analytical model. The model helps derive the system's critical parameters and indicators.
- We provide the simulation results and implement the software and hardware for model verification. The experiment can find the upper limit and compare the improved protocol with the BLE standard.
- We solve the two crucial problems of the BLE standard, namely, the insufficient support for dense deployment and exponential growth of broadcast collisions. We indicate that the new voting mechanisms are more efficient and can save more energy.

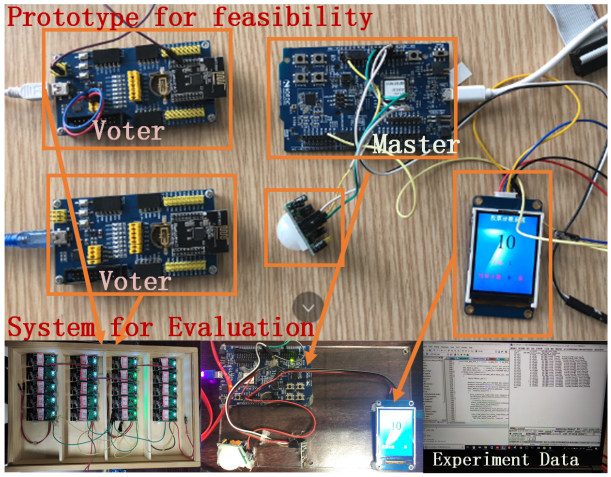


FIGURE 2. AC-UDE testbed implemented with the embedded system.

The remaining paper is organized as follows. Section II describes the motivation of this work and distinguishes it from the most relevant existing research. Sections III and IV describe the design and implementation of the active response method, respectively. Section V presents the case studies and analysis of the evaluation results. Finally, Section VI presents the concluding remarks.

II. BACKGROUND AND MOTIVATION

A. BLE AND IBEACON

According to BLE specifications, two operating modes exist for BLE devices, namely, the connectionless mode and connected mode [29]. Two types of BLE devices exist in the connectionless BLE self-organizing voting system, known as the scanner and advertiser. The advertiser (adv) periodically advertises data on an advertising channel (indices = 37, 38, 39). The scanner monitors the advertised data on the same channel to complete the interaction between the devices.

One critical advertiser parameter is the advertising interval, which consists of a fixed $advInterval$ and a pseudo-random number $advDelay$ ($0 \sim 10$ ms). The advertiser turns on the transmitter during each advertising interval and transmits the advertising data packets on the three advertising channels. The scanner scans only one fixed channel during each $scanInterval$. The opening time of the receiver is called the $scanWindow$, which cannot be longer than the $scanInterval$. When the $scanWindow$ is equal to $scanInterval$, it is called a continuous scanning mode. Both $scanInterval$ and $scanWindow$ should be less than or equal to 10.24 s.

The iBeacon protocol was first proposed and developed by Apple in 2013. The implementation of this protocol is based on the BLE advertising mode [30]. A device equipped with a BLE iBeacon is equivalent to a small base station that can send ID information specific to the iBeacon protocol through periodic advertisements. A device entering the iBeacon base station area can scan the ID information and roughly estimate the distance from the base station. Fig. 3 shows the advertising operation for an iBeacon station and a scanner that realizes passive scanning. The time sequence T related to each window event is also labeled.

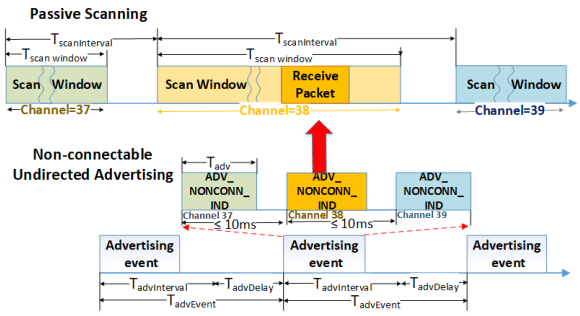


FIGURE 3. iBeacon advertising between the advertiser and scanner.

An iBeacon advertisement includes four fields: UUID (16 bytes), Major (2 bytes), Minor (2 bytes), and MPOWER (1 byte). The first three fields can be used to determine the global location of the base station and identify the base station information. MPOWER denotes the calibrated receiving power consumption at a distance of one meter from the station; this value can be used to measure the distance between the equipment and station [31]. Fig. 4 shows a BLE advertising packet format and iBeacon segment information.

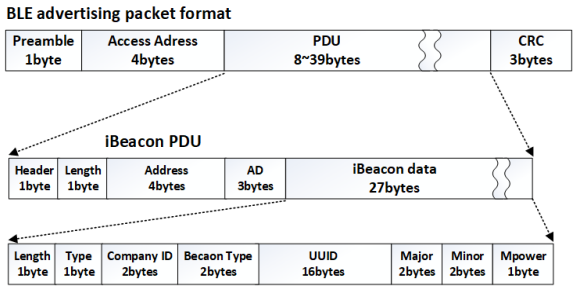


FIGURE 4. BLE package format and iBeacon PDU.

B. RELATED WORK

The BLE technology is supported by the mainstream mobile phone operating systems such as IOS and Android. Thus,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

most smartphones fully support BLE, and this widespread ecosystem can promote the popularization of BLE.

To examine the protocol format and working process of BLE and iBeacon technology, He et al. [7] designed an interactive museum system, through which visitors could acquire the navigation information and interact with collections. The intelligent display and alarm system designed by Vochin et al. [8] enhanced the display of static and dynamic information, simplified the indoor positioning of guests, and enabled the advertising and distribution of location-based content. Chen, Wu et al. [9], [10] combined iBeacon with inertial sensors to study auxiliary and precise indoor positioning and found that BLE exhibited a reduced power consumption compared to that of WiFi. Because the BLE iBeacon technology consumes less power, it can transmit signals continuously for years, using only a button cell [9]. Kriz et al. [32] located iBeacon nodes in locations in which it is difficult to provide power for a WiFi access point (AP) and the nodes are jointly located with other technologies, thereby considerably enhancing the localization accuracy. Vy et al. [11] proposed the trusted distance method to obtain the accurate locations of smartphones. A comparison of the BLE and WiFi technologies indicated that the BLE can reduce the scanning time of multiple nodes by more than 65%. Varela et al. [12] studied a service mechanism based on proximity to accurately differentiate between user groups in the same area. To exploit the BLE characteristics such as portability, energy savings, and compactness, Shao et al. [33] adopted the BLE technology to perform device-free personal identification by using a multiclassifier algorithm.

With its increasing applications, the BLE technology has been extensively investigated, leading to enhanced hardware and protocol performances. Liu et al. [23] first proposed a general model to enable device discovery on multiple channels and found that improper parameter settings between the master and slave BLE devices could significantly degrade the device discovery latency and increase the power consumption. The researchers proposed a variety of strategies based on an adaptive advertising interval to address this problem. However, this scheme increased the advertising frequency in the idle state and the device power consumption. Then Cho [34] and Julien [35] conducted a more detailed modeling analysis of the BLE device discovery process and proposed a process to optimize the master-slave device parameter selection to enhance the BLE device discovery. Bak et al. [13] proposed a BLE scanner that could simultaneously scan multiple BLE channels by using three BLE hardware frameworks to triple the theoretical scanning ability. Shan et al. [14], [15] analyzed the collisions of BLE advertising devices in dense IoT environments and proposed a model to simulate the advertising collisions. The simulation results showed that as the number of slave nodes increased, the probability of advertising collisions sharply increased, resulting in a decrease in the Bluetooth piconet throughput and an increase in the node power consumption. Furthermore, the authors also derived a new analytical model [16] to characterize the BLE discovery time considering all possible parameters. Cho [17] established a general model to perform the theoretical analysis of an M: N BLE system from the perspectives of the possibility and delay of device discovery. The model could consider the impact of the advertising channel collisions on the device discovery. However, the model implemented certain simplifications in terms of BLE advertising and scanning, resulting in a decreased accuracy. Jeon [18] proposed a general theoretical model for the neighbor discovery process (NDP). The model deduced and analyzed the delay and power consumption of the NDP based on the Chinese remainder theorem (CRT). This theoretical model and simulation results involved a certain generality that could guide the setting of voting system parameters in dense BLE environments; however, the corresponding settings cannot be applied in complex voting systems that simultaneously involve master-slave roles. Ghamari et al. [36] established a low-power Bluetooth broadcast collision model to simulate the Bluetooth broadcast collision. The simulation indicated that the throughput of the BLE network decreases rapidly with the increase in the number of broadcast collisions, and the power consumption of the nodes increases exponentially.

C. DENSE IOT ENVIRONMENTAL COLLISIONS

1) Analysis of Collision Factors

In a BLE network composed of multiple central and peripheral devices, devices work in different modes according to the required functionality at that point, e.g., advertising, scanning, and initiating. In the advertising mode, a device periodically transmits packets in the advertising channel. When two or more devices simultaneously transmit packets in the same channel, channel collision occurs, and all the packets are lost. BLE advertising packets are small (47 bytes maximum) with a maximum advertising time of 0.376 ms; however, the advertising interval is long (typical 1000 ms). Therefore, when the number of devices is small, the probability of node collision is small, and a random delay of 0–10 ms is added after each advertisement to help prevent future collisions. However, as the number of devices increases, the collision probability increases, resulting in a reduced throughput and increased power consumption [14], [15].

Another collision situation occurs during the BLE NDP. A device in the scan or initial state scans the three advertising channels sequentially. When a device in the scan state receives an ADV_IND message from the advertiser, it replies with a $SCAN\_REQ$ packet on the same channel to request additional data, while a device in the initiating state sends a $CONNECT\_REQ$ packet to initiate the connection. When multiple scanners (initiators) simultaneously send $SCAN\_REQ$ packets ($CONNECT\_REQ$ packets) to the advertiser, channel collisions occur. The collision probability continues to grow as the number of central devices increases, thereby leading to collisions in a BLE network. Although the scanner adopts a random backoff strategy to minimize the $SCAN\_REQ$ package collisions, this approach is insufficient for dense BLE environments.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

2) Existing Collision Solutions

To solve the advertising collision problem in dense BLE nodes, Seo proposed a carrier sensing scheme (CS) protocol [19] designed from the advertising node perspective by adding a 200 μs channel listening operation for each advertising channel before each advertisement. If the channel is not idle, the device must back off for some random time before advertising. This approach can effectively avoid the occurrence of advertising collisions. Next, Seo recommended the use of a dynamic advertisement interval strategy based on carrier sensing (CS-DAI) [20]. Unlike the previous work, advertisers are not required to perform any action to prevent collisions; they only need to check whether the channel is busy and dynamically adjust the advertising interval accordingly. The CS-DAI algorithm outperforms the basic CS scheme in crowded BLE networks.

Considering the network collisions caused by multiple masters responding to slave devices simultaneously, Kim [26] and L. cutrignelli [27] proposed a scan-response strategy based on a random backoff. Harris et al. [28] proposed an "opportunistic listening" algorithm that extended the capabilities of the scanning device and increased the success rate. Moreover, to solve the problem of advertising collision when the advertiser density increases, a decentralized central strategy was proposed that could aggregate multiple advertisements across similar products in a retail environment. Hernández-Solana Á et al. [37] extensively analyzed the backoff strategy for the NDP and clarified the problems and limitations of the backoff algorithm proposed in the BLE specification. To improve the efficiency of the NDP process and reduce the number of collisions, certain researchers proposed a scheme that could dynamically change the scan window size and the scan interval. To improve the scanning ability, Zhang et al. [22] dynamically adjusted the scanning interval based on two consecutive scanning results. Chen et al. [24] recorded the number of redundant devices found in each scan; a larger number of encountered redundant devices implies a more stable environment. In this study, the scan window could be enlarged, and the scan intervals could be shortened. Ng et al. [25] analyzed the device density by using a small scan interval and adjusted the scan interval more adaptively through spontaneous differential evolution. Maciej et al. [38] proposed a method to terminate the short-term broadcast operation of the broadcast node by using the scan response packet of the active scan in the large-scale voting environment, so as to reduce the broadcast collision in large-scale application scenarios.

3) Summary

The above presents a comprehensive review of BLE collision and NDP process optimization research. To avoid the occurrence of advertising collisions of the BLE edge nodes, most existing schemes adopt a dynamic advertising interval and random backoff strategy. These solutions are effective when the number of nodes is relatively small; however, these strategies cannot be adapted to dense IoT environments. The scheme based on CS requires considerably modifying the link layer of the BLE protocol. In addition, CS corresponds to increased power consumption. To solve these problems, we propose an efficient confirmation strategy based on the iBeacon protocol by using non-responding advertising and passive scanning to establish an advertising network. Although the slave nodes must perform scanning operations, which increases the power consumption, exiting the network can effectively compensate for this increase. We classified five research directions in Table 1.

TABLE 1. Summary of Research Directions

Contents	Cited Papers	Relevance
Scenarios and low energy analysis	[7]–[12], [32], [33]	common
NDP modeling	[13]–[18], [23], [34]–[36]	common
M->N multi-master broadcast	[26]–[28], [37]	weak
Master optimization methods	[22], [24]–[28], [37], [38]	common
N->1 multi-slaver broadcast	[19], [20], [38]	strong
Slaver optimization methods	[19], [20], [38]	common

In the evaluation section of this paper, we present experiments that show the performance impacts of adopting an active ACK and a random scanning time (see Section 5.1). The conclusions can be summarized as follows: (a) The random scanning strategy reduces the latency by approximately 10%, in certain cases, this strategy can avoid long-term collision problems for a very small amount by randomizing the scanning time parameter; (b) optimizing the active ACK advertising protocol has a more notable impact: this strategy can reduce the amount of repeated advertising by half, and the protocol and parameter optimizations are complementary. Furthermore, we demonstrate that combining the two methods further improves the overall performance.

III. MODEL ARCHITECTURE

A. OVERALL ARCHITECTURE

The overall architecture of the AC-UDE consists of the IoT infrastructure and cloud platform. The IoT infrastructure includes voting masters distributed among various venues and BLE voters based on the iBeacon protocol. The master obtains a voting schedule from the cloud platform and is responsible for initiating and collecting voting information. After a topic is completed, the master sends the caching results of the votes to the cloud platform within one second. Each voter performs a scan to identify the master and sends the voting information under user supervision. The cloud platform manages the IoT nodes, distributes the access IDs and keys, manages the meeting schedules, and stores the voting information. The architecture of the AC-UDE is shown in Fig. 5.

We propose an anti-collision advertising protocol to enhance the capacity and reduce latency. The next section describes the voting advertising protocol.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

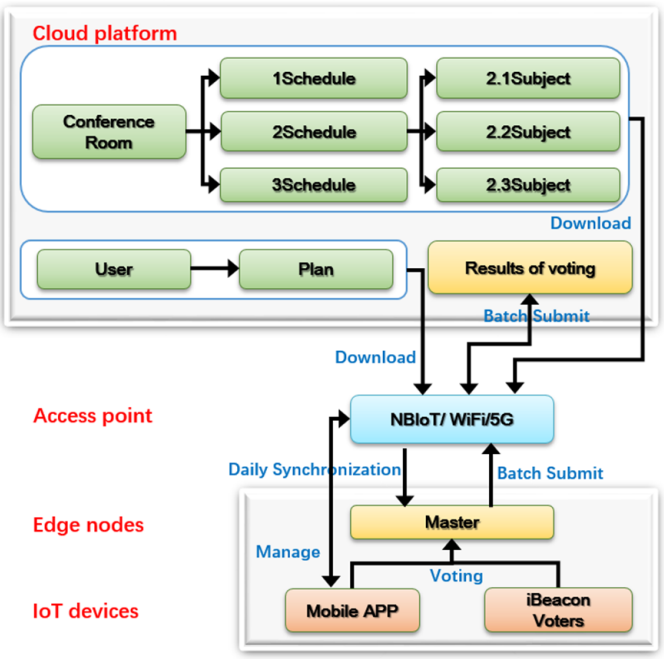
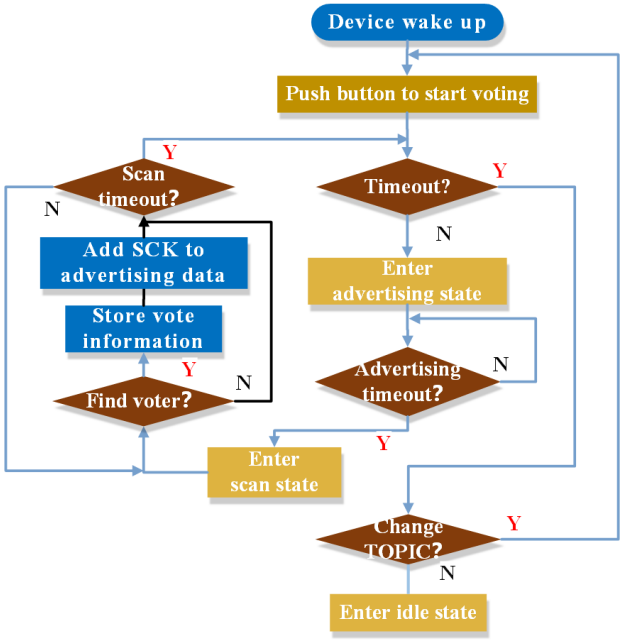
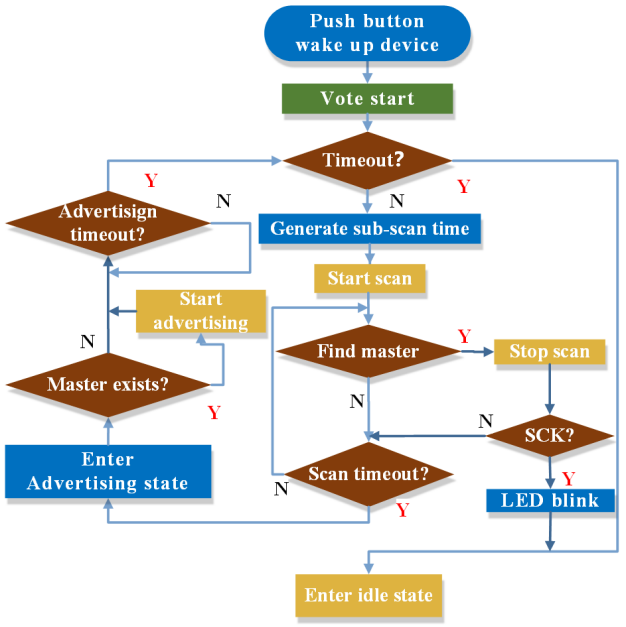


FIGURE 5. AC-UDE architecture.



(a) Process of the master.



(b) Process of the voter.

FIGURE 6. Master and voter processing.

B. ANTI-COLLISION ADVERTISING PROTOCOL

1) Protocol Framework

The anti-collision advertising protocol is based on the BLE connectionless data transmission mode and the iBeacon non-connectable undirected advertising (ADV_NONCONN_IND) packet format. The model includes a master node and M voter nodes. All the activated nodes switch between advertising and scan states at different intervals to send the voting information and responses. To extend the iBeacon definition, the protocol distinguishes the node type considering the highest bit of the major field, with one and zero corresponding to the master and voter, respectively. The last 15 bits of the major and minor fields correspond to the transmitted data. Meanwhile, the voter identifies itself through the UUID, and the master responds to the voter through the UUID.

In the model, the master broadcasts a subject. After the voter confirms the subject information, it sends a voting broadcast to the master. The master scans the voting information, registers the voter UUID, and adds certain fields to the UUID to respond to the voter when the subject is broadcast next. The master node processing is shown in Fig. 6a.

Subsequently, the voters receive the subject broadcast. When participants agree with the subject, they confirm the initiation of the voting advertising. A voter sends a vote to the master. After this voter scans the response from the master, it enters the idle state. This process is shown in Fig. 6b.

2) Response Strategy

In this protocol, after voting, the voter receives a response from the master to exit from the voting system to reduce the workload. When the number of voters is large, the master does not individually respond to the voters. Instead, based on the definition of iBeacon packets, we propose an efficient response method based on the UUID. We divide the master UUID data segments into several n bit segments (where n is a factor of 128) and load the UUID identification fields of the voters into a queue. In addition, we divide each voter UUID into a network segment field ($128 - n$ bits) and a node identification field (n bits). The network segment field distinguishes the different voting systems or geographical locations. The node identification fields differ within

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

the same network segment and can be used to distinguish different voters. Subsequently, a master can simultaneously respond to $128/n$ voters, and the same network segment can accommodate $(2^n - 1)$ different voters. The UUID formats of the master and voters are shown in Fig. 7.

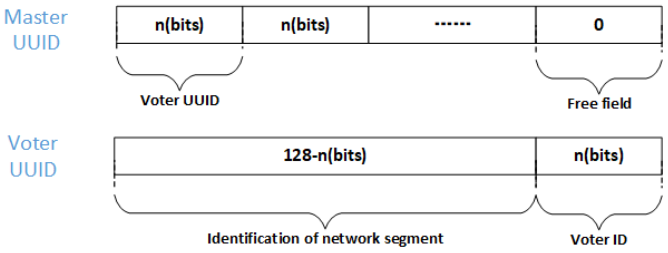


FIGURE 7. UUID formats for the master and voters.

3) Random Voter Scan

After a voter awakens, it generates a random scan time t_{ST} that is used every time the voter switches to the scan state. t_{ST} is a Gaussian distribution random variable with a standard deviation of 30 ms and a mean value of 300 ms. When the number of voters is large, multiple voters may generate the same scan time; these nodes will continue to collide each time they broadcast their votes. To avoid this aspect, when the voter enters the scanning state, a sub-scan time t generated from a uniform distribution with a mean of zero is added based on the original scanning time. The random scanning scheme is shown on the left side of Fig. 8. When voters collide under this broadcast scheme, the chance of continued collisions is reduced due to the variable scanning time randomly generated at the beginning of the scanning state, as shown on the right side of Fig. 8, $t1$ and $t2$ represent the sub-scan time generated by the voter1 and voter2, respectively.

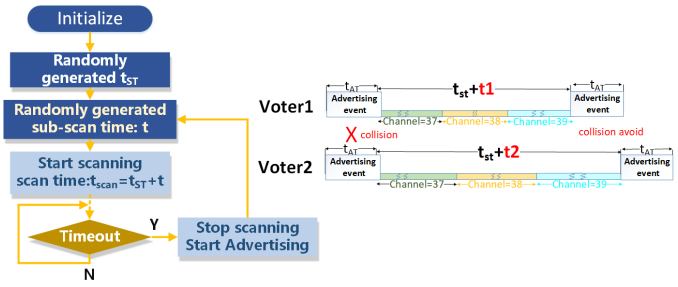


FIGURE 8. Random scan of two voters to reduce collisions.

C. ANALYTICAL MODEL

The model consists of a master node (S) and M voter nodes (V_i). The connectionless network device has two states, namely, the advertising state and scanning state. The two types of nodes in the model periodically switch between the advertising state and scanning state. During a period T, the advertising state time is t_{AT}, scanning state time is t_{ST}, scan interval is T_{SI}, and advertising time for one channel is T_{AD}. As the nodes need to complete data discovery as soon as possible, node scanning is performed in the continuous mode. The expression $\rho = t_{AT}/t_{ST}$ indicates the ratio of the advertising time to the scanning time. The superscripts S and V in these parameters refer to a master node parameter (such as t_{AT}^S) and voter node parameter (such as t_{AT}^V), respectively.

1) Voting Success Probability

To describe the conditions required for the master node to successfully discover the voting nodes, we define five events in Table 2. In the top three events, master S successfully discovers a voter V, and in the last three events, voter V successfully discovers the master S. Through these events, we can derive a formula that reflects the probability of these nodes finding one another.

TABLE 2. Events Related to the Mutual Discovery Process of Nodes

Event	Describe
E1	S and V_1 are on the same channel, S is in the scanning state, and V_1 is in the advertising state.
E2	S has sufficient time to receive V_1 broadcasts.
E3	$V_2V_3V_4\ldots\ldots$ are not on the same advertising channel as V_1 or S, that is, the advertising channel does not collide.
E4	S is on the same channel as the voter V_1, S is in the advertising state, and V_1 is in the scanning state S.
E5	V_1 has sufficient time to receive S's broadcast.

In a voting system with only M voter nodes and one master node, all the nodes independently and periodically switch between the advertising and scan state. To successfully receive a broadcast from a voter, the master must be on the same channel (i.e. 37, 38, 39) as the voter; moreover, the master must be in the scan state and have sufficient time to receive the iBeacon advertising packets from the voter. In other words, the voter must have sufficient time to send a complete iBeacon advertising packet on one of the advertising channels.

$E1$ is defined as the event in which both the master and voters are on the same channel. When the voter sends a broadcast, the probability of $E1$ is $C_3^1(1/3)^2[1/(1+\rho^S)]$, where C_3^1 represents any one of the three advertising channels, $(1/3)^2$ is the possibility of the master and voters being on the same channel, and $1/(1+\rho^S)$ is the possibility of the master being in the scanning state.

Suppose that the master receives the voter broadcast after the scanning time t_0, and the duration that the master stays on the channel is $T_{SI}^S - t_0$. The duration in which the master can receive the voter broadcast on any of the three advertising channels is identical. Thus, the possibility of $E2$, corresponding to the master having sufficient time to receive the voter broadcast, is $(T_{SI}^S - T_{AD})/T_{SI}^S$.

To ensure that the voter can be successfully scanned and found by the master, the other $M-1$ voters should

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

avoid broadcasting on this advertising channel; otherwise, channel collisions will occur, and the broadcast signal will not be received successfully. When a particular voter $V1$ broadcasts on a certain advertising channel, the probability of another voter broadcasting on the same advertising channel is $(\rho^V/(1+\rho^V))(2T_{AD}/t_{AT}^V)$, and the probability of $E3$, that is, the occurrence of communication without a collision on that advertising channel, is $[1-(\rho^V/(1+\rho^V))(2T_{AD}/t_{AT}^V)]^{M-1}$.

Finally, the probability that the master successfully receives the voter broadcast on the i-th (i = 37, 38, 39) advertising channel is the joint probability that $E1$, $E2$, and $E3$ are simultaneously true:

$$p_v(i)=\frac{1}{3}(\frac{1}{1+\rho^S})(1-\frac{T_{AD}}{T_{SI}^S})(1-\frac{\rho^V}{1+\rho^V}\frac{2T_{AD}}{t_{AT}^V})^{M-1}. \tag{1}$$

Moreover, the probability P_V of each broadcast vote being received is the sum of the probabilities of the broadcast being received on each of the three advertising channels:

$$P_V=\sum_{i=37}^{39}p_v(i). \tag{2}$$

When the master is in the advertising state, to receive advertising packets from the master successfully, the voter must be on the same advertising channel as the master and also be in the scanning state. Thus, the probability of $E4$ is $C_3^1(1/3)^2[1/(1+\rho^V)]$. In addition, the voter must have sufficient time to receive the master's advertising packets. Therefore, the probability of $E5$ is $(T_{SI}^V-T_{AD})/T_{SI}^V$. Similarly, the probability of $E3$ occurring without collision on the advertising channel is $[1-(\rho^V/(1+\rho^V))(2T_{AD}/t_{AT}^V)]^{M-1}$. The possibility that the advertising response of the master is received successfully by the voter is P_S, which is the joint probability of $E4$, $E5$, and $E3$ being simultaneously true across the three advertising channels:

$$P_S=(\frac{1}{1+\rho^V})(1-\frac{T_{AD}}{T_{SI}^S})(1-\frac{\rho^V}{1+\rho^V}\frac{2T_{AD}}{t_{AT}^V})^{M-1}. \tag{3}$$

2) Voting Latency

Voter latency is defined as the interval between the time a voter enters the active state and the time it receives the advertising packet to complete the voting. The voting process is divided into two phases, namely, a data transmission phase and a response phase. In the data transmission phase, the time $t_{ST}^V+t_{AT}^V$ represents a voter state cycle T, and its probability of success is P_SP_V. In the response phase, the average time is $0.5t_{ST}^V$, and the success rate is P_S.

The number of activated voters gradually decreases as voters complete their responses, and the complete data transmission phase of the voter must pass through $i(i\in N^+)$ complete cycles T^V plus one response. Hence, P_S and P_V change with i, where P_V^i and P_S^i are the probabilities of the completion of the data transmission and response in the $i-th$ cycle, respectively, and M_i represents the number of activated voters after the $i-th$ response. Fig. 9 shows the voter latency for completing data transmission after 1, 2, and N cycles.

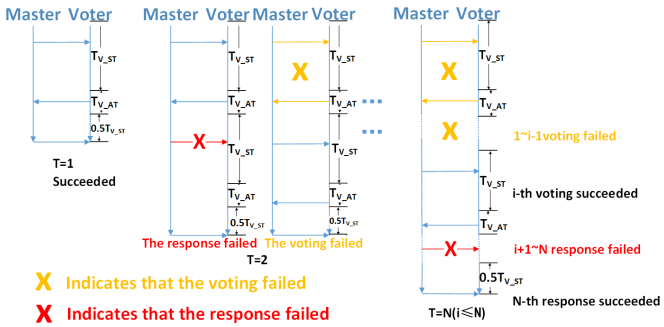


FIGURE 9. Voting latency of the data interaction.

Table 3 shows the probability that a voter receives a successful vote response from the master after $N(N=1,2...)$ cycles and the number of voters remaining on the system.

TABLE 3. Voter Data Transmission Success Probability and Latency

Latency	Remaining voters	Successful voting probability
$T^V+0.5t_{ST}^V$	$M_1=M$	$P_1=P_S^1P_V^1P_S^1$
$2T^V+0.5t_{ST}^V$	$M_2=M(1-P_1)$	$P_2=P_S^1P_V^1(1-P_S^2)+(1-P_S^1P_V^1)P_S^1P_V^2P_S^2$
...	...	...
$NT^V+0.5t_{ST}^V$	$M_N=M(1-\sum_{i=1}^{N-1}P_i)$	$P_N=\sum_{i=1}^{N}\{\prod_{j=1}^{i-1}(1-P_S^{j-1}P_V^j)\times P_S^{j-1}P_V^j\times\prod_{j=i}^{N-1}(1-P_S^j)\times P_S^N\}\ (P_S^0=P_S^1)$
...	...	...

The average latency of the voting represents the average for all the voters in a round; consequently, this value represents the average power consumption of the voters. The value can be defined as follows:

$$\pi_D=\sum_{i=1}^{\infty}P_i\times[(t_{ST}^V+t_{AT}^V)\times i+\frac{1}{2}t_{ST}^V]. \tag{4}$$

The maximum latency of the voting is defined as the period from the beginning of the voting until the time when all the voters complete voting. The formula is as follows:

$$\pi_{D_M}=(t_{ST}^V+t_{AT}^V)\times i+\frac{1}{2}t_{ST}^V \quad \{i|round(M(1-\sum_{j=1}^{i-1}P_j))==0\}. \tag{5}$$

$Round(n)$ is a rounding function that returns the integer closest to n. The maximum latency of the voting can effec-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

tively measure the maximum time required by the master to successfully collect all the votes.

3) Voting Capacity

The success rate of the voting is defined as the probability P, in that all the voters send votes to the master and successfully receive a response from the master within the time limit T_{limit}.

Within the time limit T_{limit}, the average probability of the data transmission success for each voter is p, which is the sum of the probability of the voting success p_i in each cycle.

$$p = \sum_{i=1}^{n} p_i \quad \{n = floor((T_{limit} - \frac{1}{2}t_{ST}^{V})/T^{V})\}. \tag{6}$$

Finally, the success rate P is the union of the voting success of all M voters.

$$P = p^{M}. \tag{7}$$

To satisfy the real project requirements, we limit the data transmission success rate P to a minimum value of 99.9%. Within an acceptable period of T_{limit}, the maximum number of devices that the system can carry, M_C, is calculated as follows:

$$M_C = \frac{lg0.999}{lgp}. \tag{8}$$

IV. IMPLEMENTATION DETAILS

A. SCENARIOS

In large venues, the AC-UDE implements conference process management in each room. Participants can sign in, be reminded of the schedule, conduct on-site voting, and query the results. The main steps are described as follows:

1) **Activate the system**. The moderator enters the area and turns the system on, or the system is activated by human body sensors. According to the room ID and system moderator ID, the master requests the cloud platform to obtain the conference schedule for the current date and the prompt information to start the conference.
2) **Sign in and show information for the meeting**. Participants complete the sign-in procedure based on prompts. The screen displays the meeting time, personnel, and outline of the progress for each stage.
3) **Subject voting**. The master displays the current voting subject in a loop, and the participants use buttons to vote. The system counts the number of approved votes, and the screen displays the total number of people and approved votes.
4) **Confirmation and query**. The moderator confirms the results and submits it to the cloud platform. Participants can query the results through the APP.
5) **Turn off the system**. When no processing occurs for ten minutes, the system automatically sleeps or is actively shut down by the moderator.

This process involves three confirmation requirements: sign-in, voting, and result confirmation. All these confirmation processes adopt the same flow.

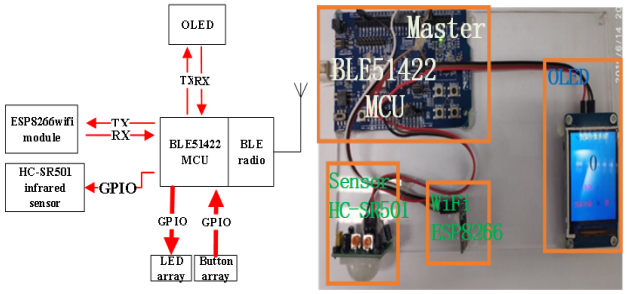
B. HARDWARE AND SOFTWARE

The system consists of three parts. Specifically, the master system, which includes the Wi-Fi module to upload the voting results to the cloud platform; and a sensor module to activate the system; an OLED module to display the interactive information; and the voter functionality embedded with the BLE module.

1) Master Hardware

The master circuit is shown in Fig. 10a: the core part is an nrf51422 BLE Micro Control Unit (MCU) [39]. The MCU is integrated with a 32-bit ARM Cortex M0 ultra-low power core and a 4.2 version of the BLE stack, which is used to build the voting network center node. The main control system embeds the OLED and esp8266wifi modules through the UART bus used to display the voting subjects and results and send this information to the cloud platform, respectively.

In addition, the master is equipped with an hc-sr501 infrared module, which monitors the external environment for information to turn the system on or off. The master has four buttons and four LEDs, which are used to turn on and off the system and display the system status, respectively. The physical diagram of the master is shown in Fig. 10b.



(a) Module design (b) Physical diagram

FIGURE 10. Hardware of master.

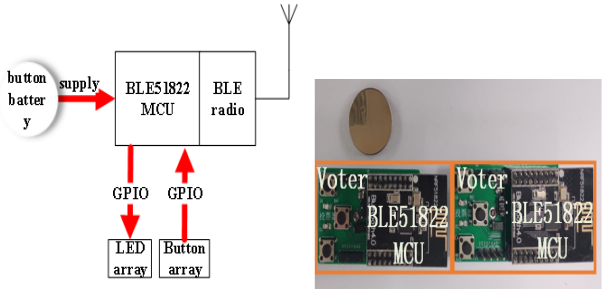
2) Voter Hardware

The voter module is shown in Fig. 11a. The core part is a BLE MCU similar to the master system, which is used to build the voting network slave nodes. To minimize the power consumption, the voter does not turn on any peripheral function. Only two buttons and LEDs are used to receive the user's voting operation and display the voting status.

When the user presses the voter button, the voter will broadcast. After receiving a voting response from the master or exceeding the voting timeout, the voter turns off the radio and enters the sleep state. Because the voting time is short (approximately 1 s), the voting peak current is small (20 mA), and the system remains in the sleep state most of the time;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

therefore, only a button cell is required as a power supply. The physical diagram of the voters is shown in Fig. 11b.



(a) Module design (b) Physical diagram

FIGURE 11. Hardware of voter.

3) Embedded Master Module

The master algorithm is presented as Algorithm 1. The master cyclically switches the broadcast and scan states during a voting period. When scanning certain votes, the ACK data segments of the voters are recorded sequentially. In the next broadcast, the subject and multiple ACK information are combined to notify the voters that have completed voting to quit in time.

The input to this algorithm is the scanning time per cycle($scanTime$), which is used to control the master scanning according to the advertising ratio. As indicated in lines 2–3, when voting is enabled, the master periodically broadcasts its information. The advertised data $advData$ contains the UUID, which includes specific system information, such as the topic, and multiple ACK data segments for multiple voters. As indicated in lines 6–10, the master enters the scanning state and records the number of votes V found during the scan, after which the master adds the required ACK information to $advData$. After the timeout, as indicated in line 15, the number of collected votes is output.

4) Embedded Voter Module

The voter algorithm is shown in Algorithm 2. The voting nodes cyclically switch between the scan and broadcast states during a voting period timeout. The node scans and considers quitting the process after receiving the ACK message, and sets a random sub-scan time each time, which reduces the probability of broadcast collision.

The input to the voter is the $scanTime$ and expected value of the scan time per cycle. At initialization, the voter randomly generates the $scanTime$ based on this average value to try and ensure inconsistent timing for each voter. When voting is enabled, as indicated in lines 2–3, the voter enters the scanning state and seeks a nearby master. As indicated in lines 4–8, if the voter finds the master, the $masterNodeExist$ flag is set as true. In the advertising state, as indicated in lines 14–15, the voter determines whether to turn on the radio by identifying the presence of a master through the $masterNodeExist$ flag. As indicated in lines 17–19, the voter completes the advertising state and generates a sub-random scan time. Finally, as indicated in lines 5–7, when the voter receives an ACK from the master during the scanning state, the voting ends; the voter blinks the led to prompt the user and enters the low power state.

Algorithm 1 Master control system.

Input: scan time per cycle, $scanTime$;
Output: number of votes, V;
Require: $V \leftarrow 0$, $t \leftarrow 0$, $state \leftarrow 0$, $advData \leftarrow uuid + topic$, $advTime \leftarrow 20$;
1: **while not** timeout **do**
2: **if** $state == 0$ **and** $t < advTime$ **then**
3: Broadcast $advData$.
4: **else if** $state == 0$ **and** $t > advTime$ **then**
5: $t \leftarrow 0$, $state \leftarrow 1$, stop Broadcast.
6: **else if** $state == 1$ **and** $t < scanTime$ **then**
7: Scan voter.
8: **if** find voter **then**
9: $advData \leftarrow advData + ACK, V \leftarrow V + 1$.
10: **end if**
11: **else if** $state == 1$ **and** $t > scanTime$ **then**
12: $t \leftarrow 0$, $state \leftarrow 0$, stop Scan.
13: **end if**
14: **end while**
15: **return** V;

C. FEATURES OF CLOUD PLATFORM

The cloud platform is based on the OneNet open IoT cloud platform, which can easily enable device access and connection functionalities such as enhanced device protocol (EDP) and Message Queuing Telemetry Transport (MQTT). The platform provides a complete solution for intelligent hardware and smart home products. On the platform, we can implement intelligent access, security management, data flow definition, and simple application functions for the voting system based on the cloud platform data flow definition.

According to the management system, the master connects to the cloud platform by establishing a TCP link using the EDP protocol, which is suitable because the master caches the results to reduce the number of submissions. The cloud platform can realize intelligent access and ensure device security by assigning access keys and unique identity IDs to specific devices. The master sends the EDP connection messages and voting information through the esp8266wifi module to the cloud platform. Four conference rooms are defined as containers corresponding to the infrastructure of the four voting systems.

Through a mobile application, which connects to and obtains the data from the cloud platform, conference rooms, schedules, and voting subjects are prefabricated. The second task involves user registration and formulation of the meeting participation plan. In addition, this framework records the user's online voting interactions, binds the user and voter devices, and processes the user's various queries.

Algorithm 2 Voting machine.

Input: average scan time per cycle, $scanTime$;
Output: none;

Require: $t \leftarrow 0$, $state \leftarrow 1$, $advData \leftarrow uuid$, $advTime \leftarrow 20$, $scanTime \leftarrow int(rand(scanTime) + 100)$, $masterNodeExist \leftarrow false$;

while not timeout **do**
  if $state == 1$ **and** $t < scanTime$ **then**
    Scan master node.
    if find master node **then**
      if ACK **then**
        LED ON, break.
      else
        $masterNodeExist \leftarrow true$.
      end if
    end if
  else if $state == 1$ **and** $t > scanTime$ **then**
    $t \leftarrow 0$, $state \leftarrow 0$, stop Scan.
  else if $state == 0$ **and** $t < advTime$ **then**
    if $masterNodeExist == true$ **then**
      Broadcast $advData$.
    end if
  else if $state == 0$ **and** $t > advTime$ **then**
    $t \leftarrow 0$, $state \leftarrow 1$, stop Broadcast.
    $scanTime \leftarrow scanTime + int(rand() * 40 - 20)$.
  end if
end while
enter low power state;
return ;

V. EVALUATION

We assessed the performance of the proposed AC-UDE considering three aspects: demonstration of the mathematical models, simulation experiments, and experiments involving physical systems. Furthermore, we also compared the proposed protocol with the iBeacon standard for collisions.

Through our simulations, we discovered the relationship between the occurrence of collisions and the three indicators:

1) Collision occurrence causes the signal to fail, thereby requiring additional repeated transmissions and increasing the latency.
2) Ultra-dense deployment, which exponentially increases the probability of continuous collision, considerably reduces the success rate.
3) Repeated broadcasts cause considerable channel occupation, increase the energy consumption, and limit the capacity to 100 nodes.

A. SIMULATION

To verify the success rate and latency of the system, we built a simulator that simulates the BLE advertising and scan process. The simulator establishes a set of state structures for each node, which represent the broadcast and scan state and operation of the node (For example, 0 and 1 represent scanning and broadcasting, respectively; the success of interaction or occurrence of channel collision is evaluated through digital comparison). The voting results are recorded at the end of each discrete-time interval. The simulator and its parameter settings are in accordance with the BLE protocol. Four types of experiments were conducted to examine the latency, success rate, advertising channel occupation, and effect of collisions on the overall performance. To ensure consistent results, we conducted 1,000 simulations for each experiment and averaged the results. The parameters are listed in Table 4.

TABLE 4. Simulation Parameter Settings

Parameter	Description	Value
M	Number of voters	$1 \sim 200$
T_{AD}	Time of transmitter on	0.5 ms
T_{WA}	Dwell time of the advertising channel	10 ms
t_{ST}^{S}	Scan time of the master	$60 \sim 300$ ms
$T_{SI}^{S} T_{SW}^{S}$	Scan interval and window of the master	$20 \sim 100$ ms
t_{AT}^{S}	Advertising time of the master	30 ms
t_{ST}^{V}	Scan time of the voter	300 ms
T_{SI}^{V}, T_{SW}^{V}	Scan interval and window of the voter	100 ms
t_{AT}^{V}	Advertising time of the voter	30 ms
T_{IFS}	Frame spacing	0.15 ms

1) Latency

Fig. 12 shows the average latency curves of the system under different parameter settings, including the number of voters M and scan time t_{ST}^{S} for the master and the voter in one cycle (the scale $t_{ST}^{S} : t_{ST}^{V}$ for the four curves in the figure decreases from 1:2 to 1:5). The graphs highlight that the results of the theoretical analysis (average number of cycles for one voter, obtained using Equation (4)) and simulation results exhibit a similar exponential trend within a large range of parameter settings. However, the curves have different slopes, and thus, the simulation can be further improved by including additional parameters. The average latency increases exponentially as the number of voters increases; moreover, the latency increases as the scale $t_{ST}^{S} : t_{ST}^{V}$ decreases from 1:2 to 1:5. These parameters affect the latency of the entire system.

Fig. 13 shows that the growth rate of the maximum latency of the voting system increases rapidly at the beginning as the number of voters increases and later gradually stabilizes. This phenomenon occurs because the number of nodes in the voting system diminishes rapidly due to the efficient response mechanism. Consequently, the occurrence probability of the voting network collisions decreases, which suppresses the maximum latency growth in the voting system. Same as Fig. 12, the optimal scale $t_{ST}^{S} : t_{ST}^{V}$ is 1:2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

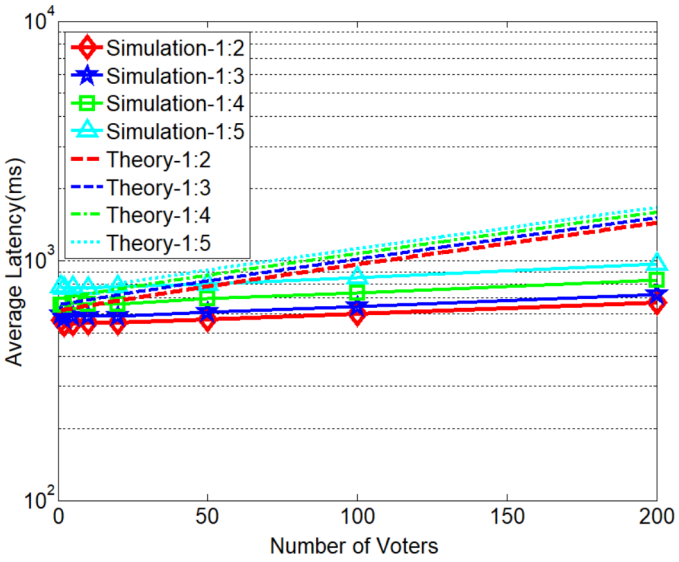


FIGURE 12. Average latency based on the number of voters and scan time.

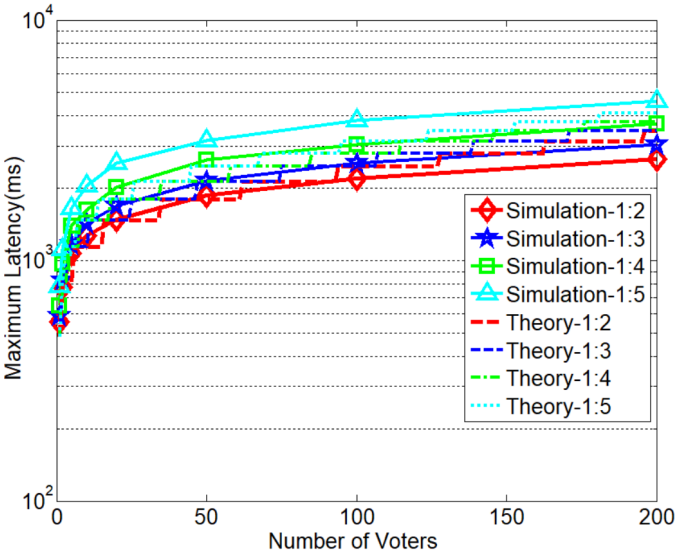


FIGURE 13. Maximum latency based on the number of voters and scan time.

2) Success Rate

In different scenarios, the data transmission is completed in different periods. We conducted simulation experiments considering three limits: 3 s, 4 s, and 5 s. Based on the conclusion of the previous experiment, the optimal scan time parameter for the master in one cycle is $t_{ST}^{S} : t_{ST}^{V} = 1 : 2$.

Fig. 14 shows the relationship between the success rate of the voting system and the number of voters under different time limits. The theoretical analysis results (voting success rate for all voters, obtained using Equation 7) are basically consistent with the simulation results in the range of 3 s, 4 s, and 5 s. As the simulation is performed using a scripting language to collect the number of collisions, the time accuracy is limited, and the results of the fitting are not highly accurate. Furthermore, the results show that the success rate increases rapidly as the time limit increases. When the time limit exceeds 5 s, the success rate can reach 99.5% for up to 200 voters. Therefore, this limiting range is more appropriate for the system response.

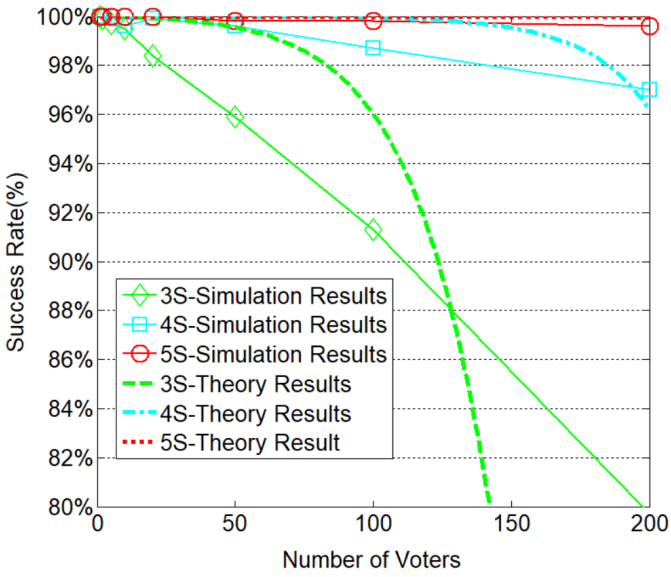


FIGURE 14. Success rate under varying time and node limits.

3) Channel Occupation

The channel occupancy chart of the voting system is shown in Fig. 15, which shows the state of the advertising channel occupancy in the simulation. The differently colored lines in the figure represent different voters; the values 37, 38, and 39 represent the node's current advertising channel, and 0 represents the scan or idle state.

Figs. 15(a) and (c) show the channel diagrams of the responding and non-responding system, respectively, when the system includes only five voters. The non-responding system can realize voting earlier than the responding system because the response signal corresponds to a higher data interaction time in the advertising channel.

In contrast, Figs. 15(b) and (d) show the channel diagrams of the responding and non-responding system, respectively, when the system includes 100 voters. In the early stages, the channel occupation is relatively dense. In the case of the responding system, the channel occupancy rate decreases sharply as the voters exit the system after receiving a response, while the non-responding system continues to exhibit its high occupancy rate. Responding to the voters can reduce the amount of repeated advertising by half.

4) Effect of Collisions

Finally, we investigate the impact of responding to the voters on the overall performance in ultra-dense environments. We adopt the previously recommended optimal parameters; specifically, the scan time ratio between the master and voter

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

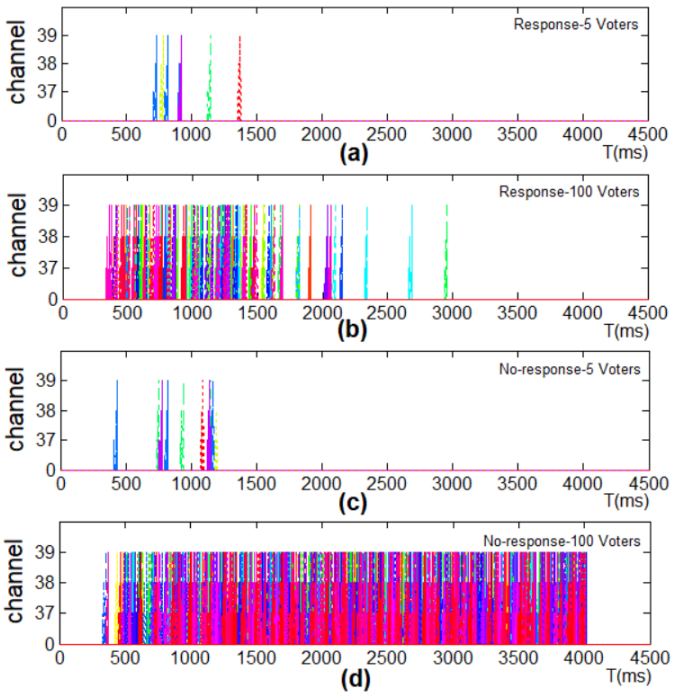
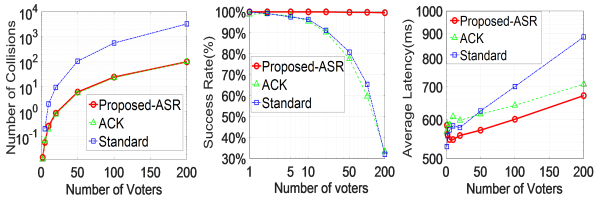


FIGURE 15. Advertising channel occupation: (a) five voters occupy the advertising channels in the responding system; (b) one hundred voters occupy the advertising channels in the responding system; (c) five voters occupy the advertising channels in the non-responding system; (d) one hundred voters occupy the advertising channels in the non-responding system.

in one cycle is $t_{ST}^{S} : t_{ST}^{V} = 1 : 2$, and the time limit is 5 s. Fig. 16 shows the performance of the three strategies, namely, responding to advertisements and scans with a random time (ASR), responding (ACK), and non-responding (standard). The figures illustrate the impact of the collisions.



(a) Number of advertising collisions (b) Voting success rate (c) Average latency of all voters

FIGURE 16. Overall performance of the three schemes.

As shown in Fig. 16(a), both the active responding schemes ACK and ASR reduce the number of collisions by more than 90%, although the impact of the ACK is higher. The data in Fig. 16(b) show that the ACK experiences continuous collision problems because it does not have a random scanning mechanism. In certain cases, a few voters always have the same scan time in experiments, resulting in long-term collisions. Therefore, the success rate is as low as that when using the standard protocol. The ASR can overcome the collision problem by involving a random scanning time. Fig. 16c shows that the ACK and ASR schemes reduce the average latency by approximately 10%.

According to Figs. 16(b) and (c), the standard exhibits a high performance when the number of voters is small. As the number of voters exceeds 20, the performance deteriorates dramatically. In contrast, the ASR performance declines gradually. Therefore, we conclude that the ASR can double the capacity. When the number of voters increases to 200, the success rate of the standard decreases to less than 40% exhibiting the same trend of increased collisions; however, the success rate of the ASR is 96% because it is not affected by the collisions.

B. VOTING SYSTEM

To verify the theory and simulation results, we consider one Nordic NRF51DK as the master and twenty NRF51822 as the voters to implement a voting system. First, we assume a simple voting environment use case. The system adopts a 120-bit segment field domain and an 8-bit voter ID flag domain. Each network segment can accommodate 255 different voters, and the master can answer 16 voters at a time. In the experiment, the number of voter nodes varies among 5, 10, 15, and 20. The average $ScanTime$ of the voters is set as 300 ms and the $ScanTime$ of the master is set as 150 ms. Furthermore, the system densely deploys nodes in a tight space, which is prone to concurrent broadcast collisions.

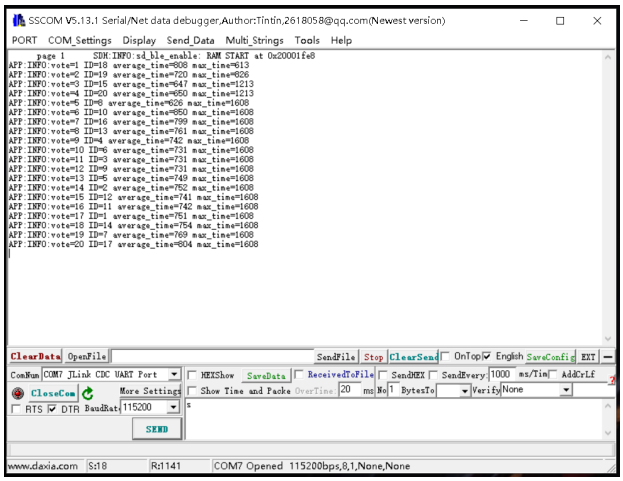


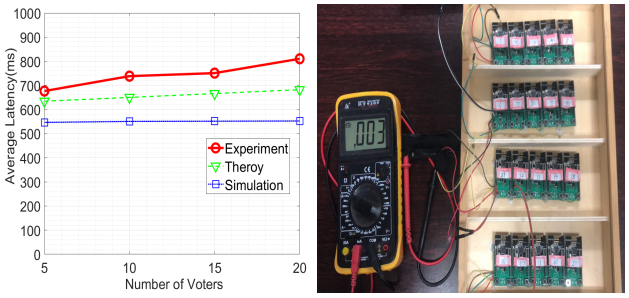
FIGURE 17. Voting information printed using a serial debugging assistant.

In addition, we increase the random wait time to simulate the thinking time as in the real world. When the voters are triggered simultaneously, they wait for a random time from 0 to 3 s. However, if the voting system is deployed in a wider area, the concurrency is more random, which is consistent with the collision effect of random broadcasting in a long time range, such as 5 s. At the end of each voting round, we allow the voter to broadcast its voting time. The master scans and records this information and prints the voting results through its serial port, and the latency is calculated. As shown in Fig. 17, we obtain all the voter IDs, average latency, and maximum latency through a serial assistant tool connected to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079120, IEEE Access

the master via USB. The tool, namely sscom, is free and can be downloaded from the URL www.daxia.com.

The experimental results are shown in Fig. 18(a). Note that the results are similar to those obtained in the simulation, as shown in Fig. 12, and theoretical models, obtained using Equation (4). The average latency of the system with 20 voters is 0.8 s. The lower latency supports low power consumption. As the static current of one voter is low when the voter is in a sleep state, it cannot be measured using the usual method. Therefore, we connect 20 voters in parallel to measure the total sleep current by using the minimum current shift of a digital multimeter. As shown in Fig. 18(b), the total reading is 3 μA.



(a) Average latency of 20 voters (b) Sleeping current of 20 voters

FIGURE 18. Performance of the actual voting system.

Fig. 19 shows the voltage waveform at both ends of the resistor during a voting broadcast. The peak current in the working state and dwell time of an advertising channel can be calculated to be approximately 26.7 mA and 2.5 ms, respectively. Because the current of the voter is low when it is in low power consumption, it cannot be measured using the same mentioned method. Thus, we measure the sleep current of the voter by using the current scale of the voltmeter. The current is less than the minimum accuracy of the voltmeter, and thus, the result is zero.

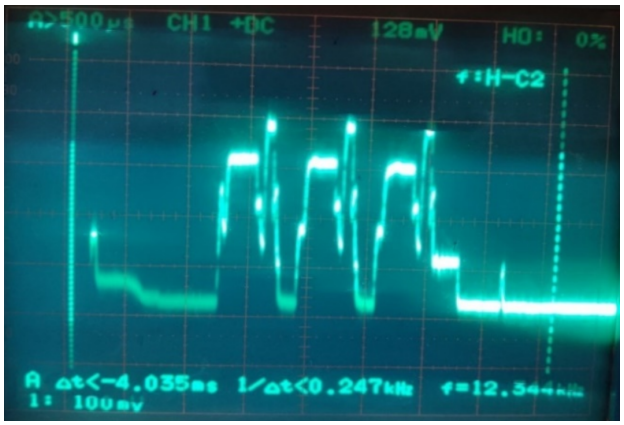


FIGURE 19. Voltage waveform of one voter.

Finally, we examine the low-power effect. When the capacity of a single button cell is 300 mAH, each voting consumed 30 mA within one second, according to the measurement. Thus, on average, votings implemented 100 times a day consume 0.8 mAH. Considering the consumption of 1.3 mAH per year in the sleep state, the cell can be used for 1–2 years.

VI. CONCLUSION

In BLE IoT networks, the schemes to reduce the signal collision can also reduce the latency and increase the capacity.

We proposed a low-collision responsive advertising protocol that can also be applied in other areas of wireless broadcast communications. The solutions include an efficient response mechanism and sub-random scan time. Furthermore, we built a mathematical model to estimate the physical capacity and latency. According to the model, we performed large-scale simulations with 200 nodes in both responding and non-responding networks. To address the two practical and open problems of the BLE standard, specifically, insufficient support for dense deployment and exponential growth of broadcast collisions, we performed experiments to identify the upper limits pertaining to 200 voters, in a 5 s time limit with an average latency of 1 s. Next, we compared the improved protocol with the BLE standard in voting scenarios. Finally, we established an efficient and energy-saving voting system based on MCU to identify the critical factors in dense IoT. The results clarify that the improved protocol with active response mechanisms outperforms the standard framework with continuous advertising.

ACKNOWLEDGMENT

This work was supported by the Key Technologies of Trusted Sharing of Multi-source and Multi-modal Data Based on Blockchain Project under Grant No.2019YFB2102403.

We thank the anonymous reviewers for their helpful comments, which helped improve the presentation of this paper.

REFERENCES

[1] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2133–2168, 2018.

[2] S. Tiwari, "An introduction to qr code technology," in *2016 International Conference on Information Technology (ICIT)*, 2016, pp. 39–44.

[3] M. Cao, L. Wang, H. Xu, D. Chen, C. Lou, N. Zhang, Y. Zhu, and Z. Qin, "Sec-d2d: A secure and lightweight d2d communication system with multiple sensors," *IEEE Access*, vol. 7, pp. 33 759–33 770, 2019.

[4] E. Ferro and F. Potorti, "Bluetooth and wi-fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[5] K. Dhivya and C. Arun, "Establishing device to device communication under cellular architecture," in *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, 2018, pp. 26–29.

[6] D. Han, X. Du, and Y. Lu, "Trustworthiness and a Zero Leakage OTMP-P2L Scheme Based on NP Problems for Edge Security Access," *SENSORS*, vol. 20, no. 8, APR 2020.

[7] Z. He, B. Cui, W. Zhou, and S. Yokoi, "A proposal of interaction system between visitor and collection in museum hall by ibeacon," in *2015 10th International Conference on Computer Science Education (ICCSE)*, 2015, pp. 427–430.

[8] M. Vochin, A. Vulpe, L. Boicescu, S. G. Obreja, and G. Suciu, "An intelligent low-power displaying system with integrated emergency alerting capability," *Sensors*, vol. 19, no. 3, 2019.

[9] Z. Chen, Q. Zhu, and Y. C. Soh, "Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1540–1549, 2016.
[10] X. Wu, R. Shen, L. Fu, X. Tian, P. Liu, and X. Wang, "ibill: Using ibeacon and inertial sensors for accurate indoor localization in large open areas," *IEEE Access*, vol. 5, pp. 14 589–14 599, 2017.
[11] T. D. Vy and Y. Shin, "ibeacon indoor localization using trusted-ranges model," *International Journal of Distributed Sensor Networks*, vol. 15, no. 1, 2019.
[12] P. M. Varela and T. Otsuki Ohtsuki, "Discovering co-located walking groups of people using ibeacon technology," *IEEE Access*, vol. 4, pp. 6591–6601, 2016.
[13] S. Bak and Y.-J. Suh, "Designing and implementing an enhanced bluetooth low energy scanner with user-level channel awareness and simultaneous channel scanning," *Ieice Transactions on Information and Systems*, vol. E102D, no. 3, pp. 640–644, 2019.
[14] Gaoyang Shan, Sun-young Im, and Byeong-hee Roh, "Optimal advinterval for ble scanning in different number of ble devices environment," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 1031–1032.
[15] G. Shan, B. Lee, S. Shin, and B. Roh, "Design and implementation of simulator for analysis of ble broadcast signal collision," in *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 448–452.
[16] G. Shan and B. Roh, "Advertisement interval to minimize discovery time of whole ble advertisers," *IEEE Access*, vol. 6, pp. 17 817–17 825, 2018.
[17] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Seo, and K. Han, "Analysis of latency performance of bluetooth low energy (ble) networks," *Sensors*, vol. 15, no. 1, pp. 59–78, 2015.
[18] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance analysis of neighbor discovery process in bluetooth low-energy networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1865–1871, 2017.
[19] J. Seo, K. Cho, W. Cho, G. Park, and K. Han, "A discovery scheme based on carrier sensing in self-organizing bluetooth low energy networks," *Journal of Network and Computer Applications*, vol. 65, pp. 72–83, 2016.
[20] J. Seo, C. Jung, B. N. Silva, and K. Han, "A dynamic advertisement interval strategy in bluetooth low energy networks," *International Journal of Sensor Networks*, vol. 27, no. 1, pp. 52–60, 2018.
[21] Y. Zhang, Z. Song, W. Wang, and Y. Tian, *An Optimized algorithm for device discovery using BLE*, ser. ACSR-Advances in Comptuer Science Research. Atlantis Press, 2016, vol. 71, pp. 115–120.
[22] J. Ch, L. Y, and Z. W, *BLE Low Energy Bluetooth Technology Development Guide*. National Defense Industry Press, 2016.
[23] Jia Liu, Canfeng Chen, and Yan Ma, "Modeling and performance analysis of device discovery in bluetooth low energy networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 1538–1543.
[24] B. Chen, S. Cheng, and J. Lin, "Energy-efficient ble device discovery for internet of things," in *2017 Fifth International Symposium on Computing and Networking (CANDAR)*, 2017, pp. 75–79.
[25] P. C. Ng, J. She, and S. Park, "High resolution beacon-based proximity detection for dense deployment," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1369–1382, 2018.
[26] J. Kim and K. Han, "Backoff scheme for crowded bluetooth low energy networks," *Iet Communications*, vol. 11, no. 4, pp. 548–557, 2017.
[27] Cutrignelli and Luca, "Channel randomization for backoff efficiency improvement," *U.S. Patent*, no. 8,583,042, Nov 2013.
[28] A. F. Harris III, V. Khanna, G. Tuncay, R. Want, and R. Kravets, "Bluetooth low energy in dense iot environments," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 30–36, 2016.
[29] B. S. I. G. (SIG), "Bluetooth core specification v4.2," p. 2772, 2014.
[30] AltBeacon, "The open and interoperable proximity beacon specification," Available online: http://altbeacon.org/, 2015.
[31] Developer, "Getting started with ibeacon," Apple, 2014.
[32] P. Kriz, F. Maly, and T. Kozel, "Improving indoor localization using bluetooth low energy beacons," *Mobile Information Systems*, vol. 2016, 2016.
[33] W. Shao, T. Nguyen, K. Qin, M. Youssef, and F. D. Salim, "Bledoorguard: A device-free person identification framework using bluetooth signals for door access," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5227–5239, 2018.
[34] K. Cho, G. Park, W. Cho, J. Seo, and K. Han, "Performance analysis of device discovery of bluetooth low energy (ble) networks," *Computer Communications*, vol. 81, pp. 72–85, 2016.
[35] C. Julien, C. Liu, A. L. Murphy, and G. P. Picco, "Blend: Practical continuous neighbor discovery for bluetooth low energy," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 105–116.
[36] M. Nikodem, M. Slabicki, and M. Bawiec, "Efficient communication scheme for bluetooth low energy in large scale applications," *Sensors*, vol. 20, no. 21, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/21/6371
[37] A. HernaNdez-Solana, D. PeRez-DiAz-De-Cerio, A. Valdovinos, and J. L. Valenzuela, "Anti-collision adaptations of ble active scanning for dense iot tracking applications," *IEEE Access*, vol. 6, pp. 53 620–53 637, 2018.
[38] M. Ghamari, E. Villeneuve, C. Soltanpur, J. Khangosstar, B. Janko, R. S. Sherratt, and W. Harwin, "Detailed examination of a packet collision model for bluetooth low energy advertising mode," *IEEE Access*, vol. 6, pp. 46 066–46 073, 2018.
[39] P. I. Andreev and B. R. Aprahamian, "Analytical comparison of bluetooth low energy beacons," in *2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA)*, 2018, pp. 1–4.

DAOQI HAN received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 1996. He is currently pursuing a Ph.D. degree at the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include intelligent IoT devices, blockchain-based access control, elliptic curve zero-knowledge proof and data privacy preservation for edge computing.

LINGYI XU Lingyi Xu received a B.S. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018. He is currently pursuing a Ph.D. degree at the School of Electronic Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include Bluetooth Low Energy technology in dense Internet of Things environments and service-function-chain construction in NFV/SDN environments.

RUOHAN CAO received the B. Eng. degree from Shandong University of Science and Technology (SDUST), Qingdao, China, in 2009, and the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. From November 2012 to August 2014, she also served as a research assistant for the Department of Electrical and Computer Engineering, University of Florida, supported by the China Scholarship Council. She is currently with the Institute of Information Photonics and Optical Communications, BUPT, as a postdoctoral researcher. Her research interests include physical-layer network coding, multiuser multiple-input-multiple-output systems, and physical-layer security.

HUI GAO received the B.Eng. degree in information engineering and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2007 and 2012, respectively. From 2009 to 2012, he was a Research Assistant with the Wireless and Mobile Communications Technology Research and Development Center, Tsinghua University, Beijing. In 2012, he was a Research Assistant with the Singapore University of Technology and Design, Singapore, where he was a Postdoctoral Researcher, from 2012 to 2014. He is currently an Associate Professor with the School of Information and Communication Engineering, BUPT. His research interests include massive and mmWave MIMO systems, UAV communications, intelligent reflecting surface aided wireless network.

YUEMING LU received the B.S. and M.S. degrees in computer science from the Xi'an University of Architecture and Technology in 1994 and 1997, respectively, and the Ph.D. degree in computer architecture from Xi'an Jiaotong University in 2000. From 2000 to 2003, he was a Researcher with the Optical Network Group Pacific, Lucent, where he was involved in 10-Gb/s optical transportation networks. He is currently a Professor with the Beijing University of Posts and Telecommunications and an Academic Committee Member with the Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education. His research interests include security control, evaluation, and data protection.