Received February 4, 2018, accepted March 12, 2018, date of publication March 19, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817343

Advertisement Interval to Minimize Discovery Time of Whole BLE Advertisers

GAOYANG SHAN AND BYEONG-HEE ROH, (Senior Member, IEEE)
Department of Computer Engineering, Ajou University, Suwon 16499, South Korea
Corresponding author: Byeong-Hee Roh (bhroh@ajou.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korean Government (MSIP) under Grant NRF-2015R1A2A2A01005577.

ABSTRACT As the Internet of Things is expected to become pervasive in everyday life, it is also expected that numerous applications and services with a large number of bluetooth low energy (BLE) devices will be developed for areas such as sports arenas, subways, shopping malls, and many other busy public places. The fast discovery of all BLE advertisers is one of the most challenging tasks in supporting such a large number of BLE-enabled services successfully. However, there has been no research published so far that presents a method of setting parameters that provide the best performance in the BLE neighbor discovery process. In this paper, we propose a method to determine an optimal value for the BLE advertisement interval to minimize the time in which all surrounding BLE advertisers are discovered by a scanner. To obtain the optimal advertisement interval, we derive a new analytical model to characterize the BLE discovery time with all possible parameters. We also analyze the effect of the optimal advertisement interval on energy consumption. The analytical model is validated by comparing the analysis results with the simulations. The simulation results show that the optimal advertisement interval for a given number of BLE advertisers can minimize the discovery time, and the energy consumption also is reduced significantly.

INDEX TERMS Bluetooth low energy, Internet of Things, neighbor discovery process, minimum discovery time, advertisement interval.

I. INTRODUCTION

With the deployment of various Internet of Things (IoT) applications, the number of physical objects connected to the Internet is growing continuously [1]. There have been intensive studies on communication and networking technologies to enable IoT applications. Examples of communication technologies include radio-frequency identifiers (RFIDs), ZigBee, Bluetooth, and Bluetooth Low Energy (BLE), and examples of networking technologies include low power and lossy networks (RPL), IPv6 over Low Power Wireless Personal Area Networks (6LowPAN), and others [2], [3]. Because the technologies have their own distinct features, their usages or communication environment may be different according to the intended application.

Recently, BLE [4], [5] has been considered as one of the most promising technologies to be adopted for IoT applications and services. BLE is a wireless protocol developed by the Bluetooth Special Interest Group (SIG) for personal area networks. BLE has been designed for low-power solutions, with sensors running on small batteries [6], suitable for various applications that require single-hop delivery of simple information such as measurement, monitoring, and control signals. BLE signals can cover a range of approximately 100 m. Owing to the simplicity, low power requirements, and low-cost features of BLE, it can be widely utilized in numerous application areas, such as sensor-enabled environmental and healthcare monitoring [7], [8], indoor localization [9], [10], smart home and automation [11], [12], tagging and proximity-based services such as near-field communication (NFC) or RFID [13], and others [14].

In BLE applications, multiple slave BLE devices (advertisers) can be associated with a master BLE device (scanner) in a star-topology piconet. The neighbor discovery process (NDP) for the association has been specified in BLE specifications 4.2 and 5.0. In NDP, each BLE advertiser sends its advertisement packets over three advertisement channels periodically, as illustrated in the next section, to its designated BLE scanner for a certain BLE-enabled service. As described in [15], BLE networks may be operated in crowded environments where a large number of advertisers may be discovered by a scanner such as in a sports arena, subway, shopping mall, and so on. As the number of advertisers increases in crowded environments, the probability of collisions also increases, and this leads to increase in power consumption and in discovery time of individual advertisers [16]. Accordingly, the fast discovery of advertisers in crowded environments becomes one of the most challenging tasks to support such BLE-enabled services successfully.

The BLE specification defines a wide range of feasible parameters and allows us to adjust them for the BLE NDP [17]. BLE parameters can be selected appropriately within their allowable ranges to support various IoT applications. There have been studies to analyze discovery time performances using simulators [18], [19], and analytical models [15], [16], [20]–[23]. In [15] and [20]–[22], analytical models for the discovery time taken by a scanner to discover a target advertiser by varying BLE parameters have been proposed. They analyzed how the discovery time is affected by the number of advertisers and the BLE parameter settings. In [23] and [24], power consumption models based on [25] have been proposed. In [23], the trade-off between the discovery time and power consumption has been analyzed. In [16], a backoff strategy of BLE devices has been proposed to reduce the collision probability and to enhance discovery latency performances during the BLE NDP. The existing studies focused on the expected discovery time and power consumption of an advertiser only, and analyzed how the discovery times are affected by varying BLE operational parameters. However, the works did not address the discovery time from the perspective where all surrounding advertisers are discovered. As described in [23], the optimal setting of BLE parameters to minimize the discovery time is essential to the development and the operation of BLE-enabled IoT services. To the best of the authors' knowledge, there has been no research on providing a method to set the parameters optimally to obtain the best overall discovery time during the BLE NDP.

In this paper, we propose a method to determine an optimal BLE advertisement interval to minimize the time needed to discover all surrounding advertisers with a scanner. It is noted that by minimizing the time for all BLE devices to be discovered, the expected discovery time for each advertiser is also minimized. The main contributions of this paper are listed as follows:

- An analytical model is derived to calculate the time in which all surrounding advertisers are discovered by a scanner. It is noted that, as mentioned earlier, the models in existing studies have focused on the discovery latency of a target advertiser.
- With the analytical model, the relationship between the discovery time of all advertisers and BLE parameter settings for a given number of advertisers is characterized. Then, we derive a method to select the optimal BLE parameter value to minimize the discovery time of all advertisers using a differential coefficient process.
- Energy consumption is analyzed to show the proposed method can reduce energy consumption of individual advertisers significantly while the discovery time is minimized.

The rest of this paper is organized as follows: the BLE NDP and related research are briefly discussed in Section II. In Section III, the analytical model for the discovery time and energy consumption of all advertisers and the calculation of the optimal advertisement interval value proposed in this paper are described. Simulation results are presented in Section IV, and the paper is concluded in Section V.

II. BACKGROUND

A. BLE NEIGHBOR DISCOVERY PROCESS

There are five possible states for the operation of a BLE link layer: standby, advertising, scanning, initiating, and connection [4]. The standby state is the default state when a device is not sending or receiving packets and the advertising and scanning states should be supported in the link layer of all BLE devices. An *advertiser* is a BLE device operating in the advertising state. Each advertiser transmits advertising messages periodically over three advertising channels indexed by 37, 38, and 39, and may expect a response from other devices. A *scanner* is a device operating in the scanning mode, and it listens to advertising messages from advertisers and may either request further information from advertisers or ignore the messages. A device in the initiating state is called an *initiator*, which listens for advertising messages and responds to them to initiate a connection with another device. The operations of initiators and scanners are identical in the BLE NDP [23].

In general, there are typically two types of advertising events: directed and undirected. The directed advertising event is used to establish connections with already known devices, while undirected advertising events are used to detect unknown devices [15]. Undirected advertising events are divided into three types: scannable, connectable, and non-connectable events [4], [5]. In the scannable and connectable undirected advertising events, a scanner or initiator may respond with additional information or request connection establishment to advertisers, respectively. On the other hand, no requests or responses from scanners or initiators are sent in the non-connectable undirected advertising events. In this paper, we focus on the non-connectable undirected advertising events, but normally there is no significant difference for other undirected advertising events in the BLE NDP.

For all undirected or directed advertising events, the advertiser generates advertising events (*advEvent*) periodically, and the time between the start of two consecutive advertising events (*T_advEvent*) is composed of two parts: a fixed advertisement interval, *advInterval* (R), and a pseudo-random delay, *advDelay* (T_d). Each *advInterval* consists of a sequence of three advertising messages (ADV_PDU, shown in Fig. 1, and noted as ADV_IND PDU in BLE specification [4], [5]) transmitted over three advertising channels indexed by 37, 38, and 39, respectively, as shown in Fig. 1. According to the BLE specification [4], the *advInterval* should be an integer multiple of 0.625 ms in the range

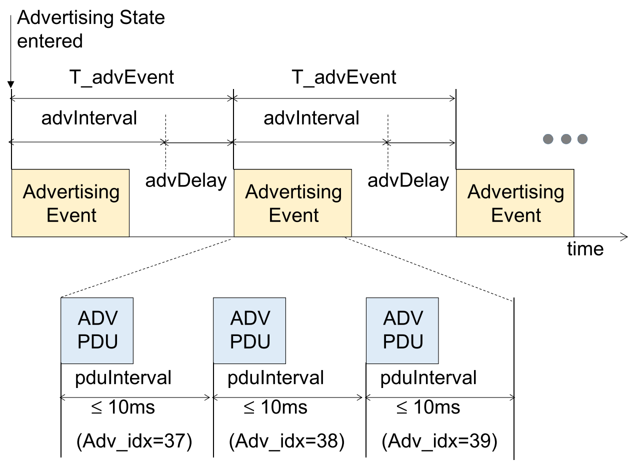


FIGURE 1. BLE advertising process.

of 20 ms to 10.24 s. In BLE specification 5.0 [5], the range of *advInterval* is extended to 10485.759375s. If the advertising event type is either a scannable undirected or a non-connectable undirected event type, the *advInterval* shall not be less than 100 ms. The *advDelay* should be within the range of 0-10 ms. The time between two consecutive ADV_PDUs, *pduInterval*, is at most 10 ms. The minimum and maximum ADV_PDU sizes are 8 and 47 bytes, respectively. Let T_p be the transmission time for an ADV_PDU. When we consider the reference BLE signal bit rate of 1 Mbps and the maximum ADV_PDU size of 47 bytes, the maximum T_p is 0.376 ms.

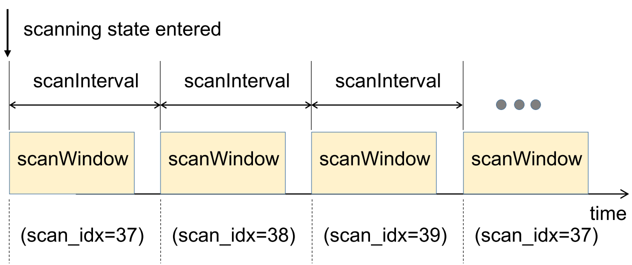


FIGURE 2. BLE scanning process.

There are two types of scanning mode: passive and active. In passive scanning mode, the scanner only receives ADV_PDUs from advertisers, but it does not send any packets to them. In active scanning, the scanner listens for ADV_PDUs and may send requests for additional information to advertisers depending on the ADV_PDU types. The BLE scanning process is shown in Fig. 2. The scanner periodically scans an advertising channel in the sequence of channel indexes 37, 38, and 39 in a round-robin manner. The duration for scanning each channel is defined as a scan window, *scanWindow* (T_w), and the interval between two consecutive scan windows is defined as a scan interval, *scanInterval* (T_s). According to the BLE specification 4.2 [4], the *scanWindow* and *scanInterval* should be less than or equal to 10.24 s. In BLE specification 5.0 [5], the range of *scanWindow* and *scanInterval* are extended to 40.96s. The *scanWindow* should be less than or equal to *scanInterval*.

When the *scanWindow* and the *scanInterval* are set to the same value, the scanner should scan each advertising channel continuously without sleeping, referred to as continuous scanning mode. On the other hand, in the discontinuous scanning mode, the *scanWindow* is less than the *scanInterval*, accordingly the scanner alternatively scans the channel during *scanWindow* and sleeps for the remaining time in every *scanInterval*.

B. RELATED WORKS

In the BLE NDP, parameter settings affect the discovery latency performance as discussed in [15], [20], [21], and [22]. An analytical model based on a pure ALOHA system model for a continuous scanning mode has been introduced [15]. With the analytical model, the authors showed that the number of advertisers directly affects the average discovery latency, that is, as the number of advertisers increases, the average latency time in which an advertiser is discovered by a scanner also increases exponentially. In [20], it has been shown how the ratios of *T_advEvent* and *scanWindow* to *scanInterval* affect the average discovery time of an advertiser. Liu et al. [21] derived a performance model to analyze the average discovery time by varying scanning and advertisement intervals for the discontinuous scanning mode, revealed some potential pitfalls of BLE, and proposed a method to adjust the advertisement interval to solve the problem of long discovery latency by scanners. Cho et al. [22] proposed an analytical model to calculate the discovery probability, and investigated how BLE discovery time performances are influenced by BLE parameter settings. Most of existing literatures have focused on BLE specification 4.2 [4], since BLE specification 5.0 [5] has been recently announced. Two types of NDPs are defined in BLE 5.0: basic and advanced NDPs. The operation of the basic NDP in BLE 5.0 is almost same as the NDP in BLE 4.2 except for the allowable range of BLE parameters. In [26], the basic NDP performances for BLE 5.0 has been analyzed.

Lowering energy consumption during the BLE NDP is also one of the most important key design requirements for BLE-enabled IoT applications. Energy consumption affects the lifetime of BLE devices directly. In [25], a power consumption model was proposed by capturing and analyzing current waveforms from an advertiser during the NDP. Jeon et al. [23] proposed a performance model to evaluate the trade-off between the BLE discovery time and the energy consumption. In the model, only one advertiser is tagged as an advertiser, and all other advertisers are considered sources of interference. They investigated the interaction of *advInterval*, *scanWindow*, and *scanInterval* on the discovery delays and on power consumption. Schrader et al. [24] derived a power consumption model for the BLE NDP to predict the impact of BLE advertisement parameters on long-term average power consumption. In [16], a backoff strategy for BLE devices to reduce the collision probability was proposed to enhance the discovery time and the power consumption performance.

III. OPTIMAL ADVERTISEMENT INTERVAL FOR MINIMUM DISCOVERY TIME

In this section, we investigate an analytical model to calculate the total time for advertisers to be discovered by a scanner. We consider the case where there is one scanner, and N advertisers that reside in the scanner's discovery range. We also consider the undirected advertising events for advertisers and the passive continuous scanning mode for the scanner. As mentioned in Section II, there is no significant difference for other advertising events and scanning modes in the BLE NDP. That is, each advertiser's advertising process, shown in Fig. 1, is independent of the others, and it competes with others fairly for discovery by the scanner. Then, with the analytical model, we present a method to determine the advertisement interval to minimize the discovery time. The terms utilized in the analysis are given in Table 1.

TABLE 1. Terms used for the analysis.

Symbol	Description
N	number of BLE advertisers to be discovered
R	advertisement interval parameter ($advInterval$)
T_d	pseudo-random delay ($advDelay$)
T_s	time duration for a scan interval ($scanInterval$)
T_w	time duration for a scan window ($scanWindow$)
T_p	transmission time for a packet
T_{pi}	$pduInterval$ shown in Fig. 1
p_{coll}	probability that an ADV_PDU collides with another
p_{succ}	probability that an ADV_PDU is successfully received by the scanner
p_{fail}	probability that an ADV_PDU fails to be discovered by the scanner
$\bar{t}_{pdu}$	mean time interval between two consecutive ADV_PDUs
$\bar{t}_{scan}$	mean time interval between two consecutive ADV_PDUs successfully received at the scanner
C_{scan}	expected number of ADV_PDUs captured by the scanner until all surrounding advertisers are discovered
$T_{discover}$	expected time in which a scanner discovers all surrounding advertisers
E_1	energy consumption of an advertiser during $T_{discover}$

When a specific advertiser sends an ADV_PDU, if any part of the ADV_PDU from another advertiser overlaps with the transmission time of the specific advertiser's ADV_PDU over $2 \times T_p$, they collide with each other. So, similarly to [23], the probability that an ADV_PDU collides with another, p_{coll}, is

$$p_{coll} = \frac{2T_p}{R + E[T_d]} \quad (1)$$

where $E[T_d]$ is the mean of the pseudo-random delay.

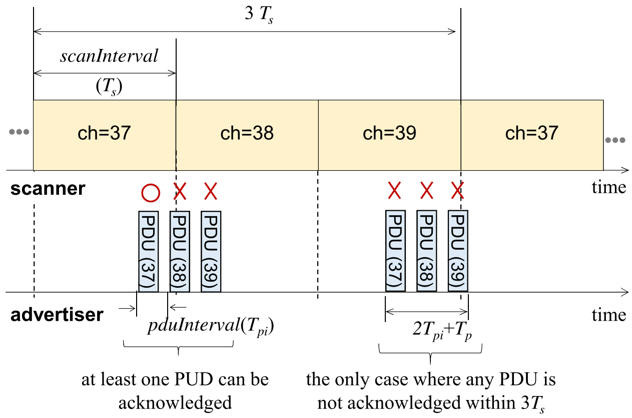


FIGURE 3. Example case where a scanner cannot acknowledge ADV_PDUs from advertisers.

As described in Section II, advertisers and the scanner utilize channel indexes 37, 38, and 39 periodically in a round-robin manner. The length of a *scanInterval* should be larger than that of an *advInterval*. So, an ADV_PDU can be successfully discovered by the scanner only when it is completely received without loss and the advertiser's and the scanner's channels coincide with each other. Though it is successfully received at the scanner, if the advertiser's channel is different from the channel being scanned at the receiving time, it cannot be acknowledged. Fig. 3 shows the case where the scanner can acknowledge ADV_PDUs from advertisers or not within three consecutive *scanInterval*s. The first ADV_PDU from channel 37 can be acknowledged by the scanner as their channel indexes are the same. The second and the last ADV_PDUs cannot be acknowledged because the scanner changes its scanning channel during the ADV_PDUs' transmission time, and some bits are lost. The following 3-rd, 4-th, and 5-th ADV_PDUs also cannot be acknowledged, even though the scanner does not change its scanning channel during their transmission time, because the advertising and the scanning channels are different. Because each advertiser operates independently, the probability that the scanner successfully receives a complete message from any advertiser is

$$p_{succ} = \left(1 - \frac{2T_{pi} + T_p}{3T_s}\right) \times (1 - p_{coll})^{(N-1)} \quad (2)$$

where T_{pi} denotes the *pduInterval* shown in Fig. 1.

Because each advertiser independently sends ADV_PDUs periodically, the expected time interval between two consecutive ADV_PDUs from all N advertisers is

$$\bar{t}_{pdu} = \frac{R + E[T_d]}{N}. \quad (3)$$

Although ADV_PDUs are sent from advertisers with the average interval given by Eq. (3), they may be lost due to collision. Let $\bar{t}_{scan}$ be the expected time interval between two consecutive ADV_PDUs successfully received by the scanner. Then, we have

$$\bar{t}_{scan} = \frac{\bar{t}_{pdu}}{p_{succ}} \tag{4}$$

With Eqs. (2), the probability that the scanner unsuccessfully receives an ADV_PDU can be computed as follows:

$$p_{fail} = 1 - p_{succ}. \tag{5}$$

Assume that the time is divided into a constant interval of Δ $(= R+E[T_d])$ once the scanner has entered the scanning state. Let us also assume that all advertisers simultaneously enter the advertising state at the start time of the scanning state. Then, in every Δ interval, all N advertisers send ADV_PDUs once as shown in Fig. 4. Each number j $(j = 1, 2, \ldots, N)$ in every Δ interval on the advertisers line in Fig. 4 indicates the j-th generated ADV_PDU during the interval not generated by the j-th advertiser. With the pseudo-random delay (*advDelay*), the order of advertisers can be changed at every Δ interval. According to the BLE NDP explained in Section II, each ADV_PDU can be received by the scanner with a probability p_{succ}, and each advertiser may be discovered multiple times by the scanner before all advertisers are completely discovered.

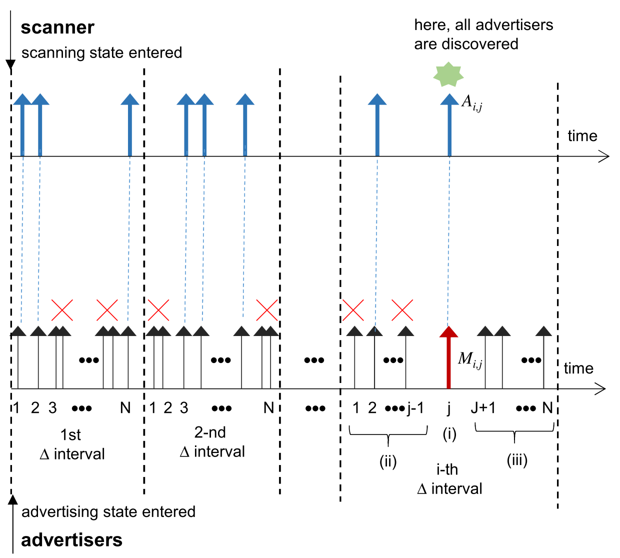


FIGURE 4. Scanning process to discover all advertisers.

Let $M_{i,m}$ and $A_{i,m}$, where $i = 1, 2, \ldots$, and $m = 1, 2, \ldots, N$, be the m-th generated ADV_PDU during the i-th Δ interval and the advertiser that generates $M_{i,m}$, respectively. In addition, let $P_{i,m}$ $(i = 1, 2, \ldots$ and $m = 1, 2, \ldots, N)$ be the probability that $M_{i,m}$ is received successfully by the scanner. As mentioned earlier, advertisers are detected multiple times by the scanner until all advertisers are discovered. Consider the case where the scanner discovers all advertisers upon receiving $M_{i,j}$. This indicates that advertiser $A_{i,j}$ has not been discovered by the scanner previously, and it is the last N-th discovered advertiser. That is, all N advertisers are completely discovered by the scanner upon receiving $M_{i,j}$. To calculate $P_{i,j}$, which is the probability that the scanner discovers all advertisers at j-th ADV_PDU during the i-th Δ interval, we consider the three cases: (i), (ii), and (iii) shown in Fig. 4.

In case (i), all the previous ADV_PDUs sent by $A_{i,j}$ up to the $(i-1)$-th Δ interval have been lost due to collisions, and only $M_{i,j}$ is successfully delivered to the scanner. Then, the probability of case (i) is given by $p_{fail}^{i-1} \times (1 - p_{fail})$.

Case (ii) corresponds to $(j-1)$ $M_{i,k}$s where $k = 1, 2, \ldots, j-1$. The scanner received at least one ADV_PDU from $A_{i,k}$ $(k = 1, 2, \ldots, j-1)$ successfully before $M_{i,j}$. In this case, each $A_{i,k}$ $(k = 1, 2, \ldots, j-1)$ sent i ADV_PDUs once every Δ interval before $M_{i,j}$, and at least one ADV_PDU among them should be successfully delivered to the scanner. Because the $j-1$ advertisers operate their advertising events independently, the probability for this case is $(1 - p_{fail}^{i})^{j-1}$.

Case (iii) corresponds to $(N$-$j)$ $M_{i,n}$s where $n = j+1, j+2, \ldots, N$. In case (iii), at least one ADV_PDU from $A_{i,n}$ $(n = j+1, j+2, \ldots, N)$ has been successfully received by the scanner before the i-th Δ interval. Because each $A_{i,n}$ sent $(i-1)$ ADV_PDUs once every Δ interval before the i-th Δ interval, the probability for case (iii) is $(1 - p_{fail}^{i-1})^{N-j}$.

To cause $A_{i,j}$ to be the last discovered N-th advertiser, cases (i), (ii), and (iii) should be met simultaneously. Then, we have

$$P_{i,j} = p_{fail}^{i-1}(1 - p_{fail})(1 - p_{fail}^{i})^{j-1}(1 - p_{fail}^{i-1})^{N-j}. \tag{6}$$

To provide the validation of Eq. (6) intuitively, we consider the probability that all N advertisers are discovered during the first Δ interval. Because all ADV_PDUs from N advertisers should be transmitted to the scanner without collision, the probability is written as $(1 - p_{fail})^N$. We can obtain the same probability from Eq. (6) for $P_{1,N} = (1 - p_{fail})^N$.

Let C_{scan} be the expected number of ADV_PDUs successfully captured by the scanner until all surrounding advertisers are discovered. Then, we have

$$C_{scan} = (1 - p_{fail}) \times \left\{ N(1 - p_{fail})^N + \sum_{i=2}^{\infty} \sum_{j=1}^{N} ((i-1)N + j)P_{i,j} \right\}. \tag{7}$$

We can obtain the expected time for the scanner to discover all surrounding advertisers, denoted by $T_{discover}$, using

$$T_{discover} = \bar{t}_{scan} \times C_{scan} \tag{8}$$

Proposition 1: There exists an optimal *advInterval* value to minimize $T_{discover}$.

Proof: The first derivative of the partial differential equation for $T_{discover}$ with respect to R is

$$\frac{\partial T_{discover}}{\partial R} = \frac{\partial \bar{t}_{scan}}{\partial R} \times C_{scan} + \bar{t}_{scan} \times \frac{\partial C_{scan}}{\partial R}. \tag{9}$$

Because it is difficult to derive Eq. (9) as a single form, we utilize a numerical approach to calculate Eq. (9). Let R_{min}

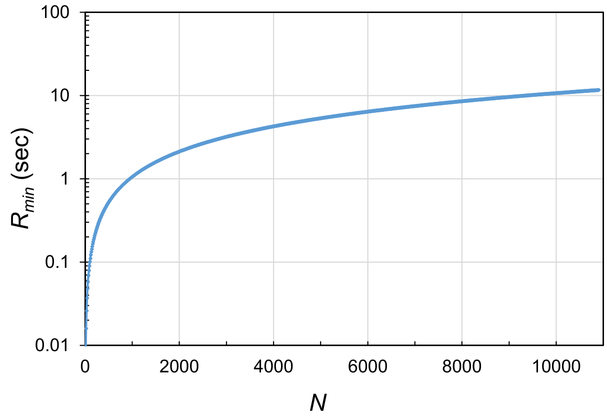


FIGURE 5. R_{min} **varying** N.

be the *advInterval* value to make $\partial T_{discover}/\partial R$ be zero. That is, $\partial T_{discover}/\partial R|_{R=R_{min}} = 0$. Fig. 5 shows the R_{min} obtained by varying the number of advertisers (N). As shown in Fig. 5, we can see that R_{min} exists for every N, and it increases as N increases.

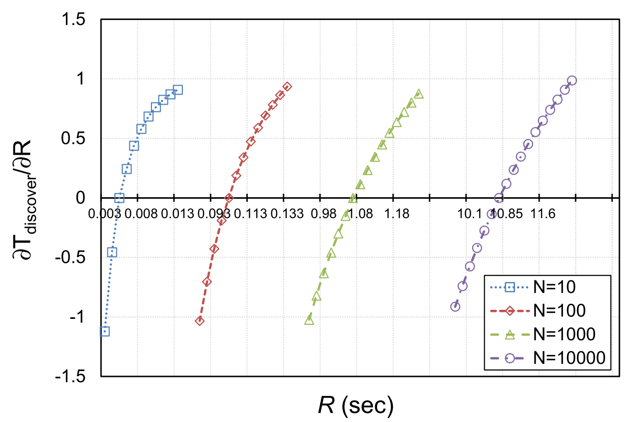


FIGURE 6. **Plots of** $\partial T_{discover}/\partial R$ **according to** R **for various** N.

Fig. 6 shows the variation of $\partial T_{discover}/\partial R$ according to the variation of R for $N = 10$, 100, 1000, and 10,000. We can see from Fig. 6 that $\partial T_{discover}/\partial R < 0$ when $R < R_{min}$, $\partial T_{discover}/\partial R > 0$ when $R > R_{min}$, and $\partial T_{discover}/\partial R = 0$ when $R = R_{min}$ for all N. In other words, we have the relationship among $\partial T_{discover}/\partial R$, R, and R_{min} as follows:

$$sign(\frac{\partial T_{discover}}{\partial R}) = \begin{cases} -, & R < R_{min} \\ 0, & R = R_{min} \\ +, & R > R_{min}. \end{cases} \tag{10}$$

The typical shape of a curve satisfying the relationship of Eq. (10) is convex. So, $T_{disocver}$ has its minimum value at R_{min}. ■

Proposition 2: When $R_{min} \leq 0.02s$ for a given N, then the *advInterval* value to minimize $T_{discover}$, R_{min}, is 0.02s.

Proof: According to the BLE specification, R should be an integer multiple of 0.625ms in the range of 0.02s to 10.24s. So, when $R_{min} \leq 0.02s$, $T_{discover}$ is always increasing as R increases. Accordingly, we can obtain the minimum $T_{discover}$ at 0.02s. ■

Proposition 3: When $R_{min} \geq 10.24s$ for a given N, then the *advInterval* value to minimize $T_{discover}$, R_{min}, is 10.24s.

Proof: From Eq. (10), when $R_{min} \geq 10.24s$, $T_{discover}$ is always decreasing as R increases. Accordingly, we can obtain the minimum $T_{discover}$ at 10.24s. ■

From Propositions 1-3, the algorithm to quickly compute R_{min}, the *advInterval* value to minimize the time that the scanner discovers all advertisers, is given in Fig. 7, where δ is 0 or a very small value.

```
N is given; let define f(x)= ∂T_discover/∂R |_{R=x};
R_low = 0.02; R_high = 10.24;
if (f(R_low) > 0 and f(R_high) > 0 ) then
   R_min = R_low;          // by Proposition 2
elseif (f(R_low) < 0 and f(R_high) < 0 ) then
   R_min = R_high;         // by Proposition 3
else
   while ( 1 )
     R_mid = (R_low + R_hihg)/2;
     if ( f(R_mid) ≤ δ) ) then
       R_min = R_mid; // by Proposition 1
       return;
     elseif (sign(f(R_low))) = sign(f(R_mid))) then
       R_low = R_mid;
     else // sign((R_high)) = sign(f(R_mid))
       R_high = R_mid;
     endif
   endwhile
endif
```

FIGURE 7. **Algorithm to compute** R_{min}.

From the BLE specification [4], [5], the advertisement interval R should be an integer multiple of 0.625 ms. Because the R_{min} value obtained from Propositions 1, 2, and 3 can be a real number, we finally choose the calculated R_{min} value as the closest value which is an integer multiple of 0.625 ms to meet the BLE specification, which can be defined as

$$R_{opt} = Round(\frac{R_{min}}{\theta}) \times \theta \tag{11}$$

where $Round(x)$ is the nearst integer to x, and θ is 0.625×10^{-3}s.

To calculate the energy consumption of an advertiser during $T_{discover}$, we assume that the operational voltage of an advertiser is constant, and the energy consumption is defined as the electric current multiplied by the elapsed time, similar to [23]. Let e_a be the energy consumption of an advertiser during a single advertisement period, i.e. $T\_advEvent$. Then, e_a can be calculated using the energy consumption models in [23] or [25]. Then the expected energy consumption of an advertiser during $T_{discover}$ is

$$E_1 = \frac{T_{discover}}{R + E[T_d]} \times e_a \tag{12}$$

Note that the total energy consumption for all N advertisers during $T_{discover}$ is simply given as $E_N = E_1 \times N$.

IV. NUMERICAL RESULTS

To validate the analytical model, we conducted extensive simulations by developing a BLE simulator extended from [18]. The system parameters of advertisers and the scanner for the analysis and the simulations are given in Table 2. As mentioned previously, the continuous scanning mode is used, i.e. $scanInterval = scanWindow$.

TABLE 2. System parameters for the experiments.

Parameters	VALUE
$E[T_d]$	5 ms
T_p	0.376 ms
T_s	10.24 s
$pduInterval$	10 ms
Tx-power	4dB

Fig. 8 shows $T_{discover}$ values for different R and N. For a given R, as N increases, $T_{discover}$ increases. From Fig. 8, there exists a R value that minimizes $T_{discover}$ for each N, which is denoted as R_{opt}, and it varies for different N. For N larger than 100, $T_{discover}$ values with short R are much longer than those with long R. On the other hand, when N are less than 100, $T_{discover}$ values with short R are shorter than those with long R.

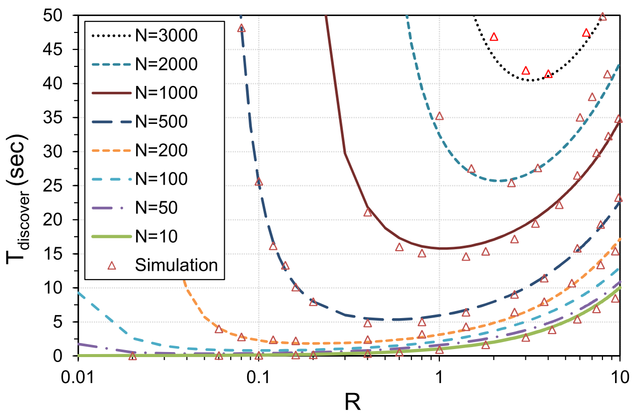


FIGURE 8. $T_{discover}$ **values obtained by varying** R **for different** N**.**

Fig. 9(a) compares $T_{discover}$ values with different R by varying N from 10 to 5,000. In general, most advertisers are set to approximately 1s by default. As we can see from Fig. 9(a), $T_{discover}$ values with $R = R_{opt}$ achieve their minimums for all N compared with those with R set arbitrarily from 20ms to 10.24s, which is the allowable range specified in BLE 4.2 [4]. Compare the line when $R = 0.2$s with the line when $R = 10.24$s, we can find when N is smaller than 800, the shorter R is the shorter $T_{discover}$ is. This is because when N is small, the number of signals from advertisers is low enough to avoid many collisions. In contrast, when N is larger

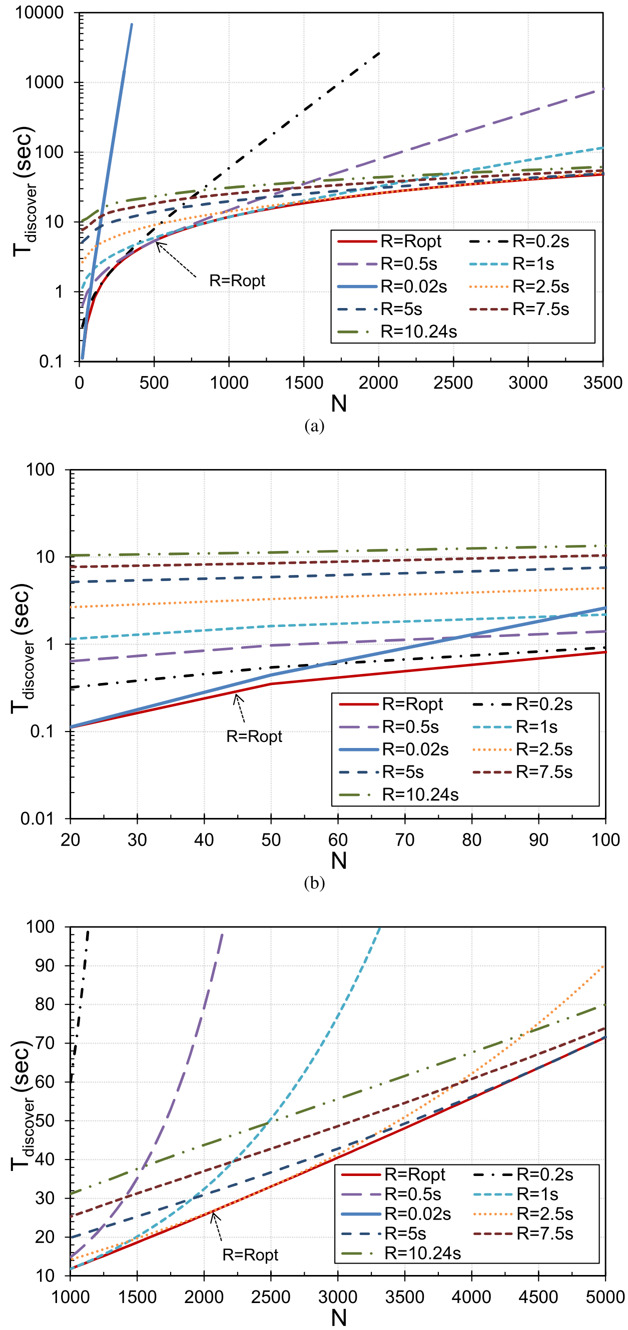


FIGURE 9. **Effects of** R **on** $T_{discover}$**: (a) for all** N **values, (b) for small** N **values, (c) for large** N **values.**

than 800, as shown in Fig. 9(a), a shorter R causes $T_{discover}$ to be longer. This is because when N is very large, the shorter R is, the higher the collision probability among ADV PDUs is. In this case, longer R can mitigate signal collision rates.

Especially, Fig. 9(b) and Fig. 9(c) show the variation of $T_{discover}$ for small N less than 100 and for large N greater than 1000, respectively. For small N as shown in Fig. 9(b), we can see that as R becomes shorter, $T_{discover}$ is lower. On the other hand, the shorter R causes $T_{discover}$ to be much longer for large N as shown in Fig. 9(c). For all N, whether they are small or large in quantity, $T_{discover}$ values with $R = R_{opt}$ remain at their lowest levels.

With the results of Fig. 9, the proposed method can be utilized for designing applications sensitive to $T_{discover}$. For example, an inventory management system may require a short delay to discover all goods in a warehouse. When the requirement of $T_{discover}$ for an application scenario is given, then the maximum N to meet the requirement can be easily computed using the proposed method. As we can see from Fig. 9, $T_{discover}$ > 10s for $N > 1000$, and $T_{discover}$ < 5s for $N < 500$. Then, if the requirement of $T_{discover}$ for a certain service should be below 5s, the number of advertisers that a scanner can manage effectively should be less than 500.

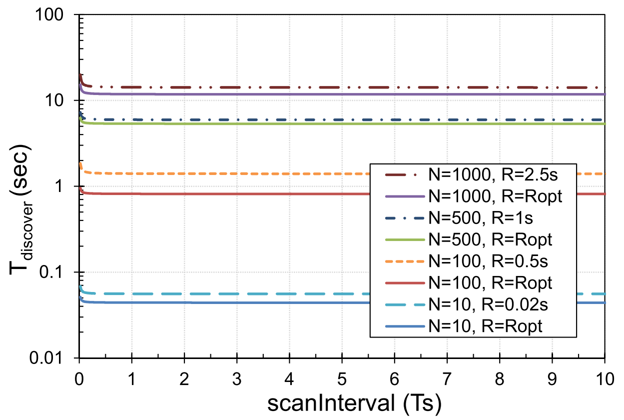


FIGURE 10. Effects of *scanInterval* on the $T_{discover}$.

Until now, we have investigated the effect of R on the discovery time for a given N. Along with R, the *scanInterval* (T_s) of the scanner is also a very important parameter for the operations of BLE-enabled services. Fig. 10 shows the effect of the variation of the *scanInterval* on $T_{discover}$ for different N and R values. When *scanInterval* < 0.05 s, $T_{discover}$ values for a given N are affected by the variation of *scanInterval*. However, for *scanInterval* > 0.1 s, $T_{discover}$ values decrease very slightly as *scanInterval* increases, but they appear to remain at a certain constant level. The result indicates that *scanInterval* does not affect $T_{discover}$ because *scanInterval* is usually set to larger than 1 s. From all simulation results, because $T_{discover}$ values are minimum at $R = R_{opt}$ for all *scanInterval* variations for each N, we can achieve the minimum $T_{discover}$ by setting R to R_{opt}.

Fig. 11 shows the expected energy consumption of an advertiser during $T_{discover}$ by varying R for different N. For the calculation of the energy consumption, we used the model proposed in [23]. With this model, the energy consumption of an advertiser during one advertisement interval is given as 532.034 (mA*ms). The energy consumption at $R = R_{opt}$ for each N is denoted by a circle in Fig. 11. From Fig. 11, for each N, energy consumption of each advertiser is maximum at R = 20ms, and decreases significantly until $R = R_{opt}$. This is because long R can mitigate signal

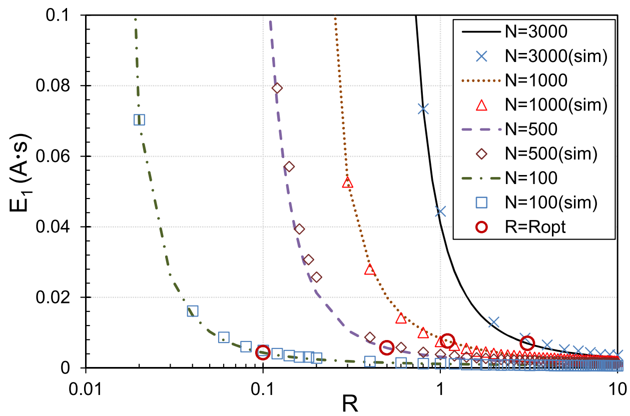


FIGURE 11. Energy consumption for an advertiser during $T_{discover}$ for different N.

collisions for large N as described in Fig. 9(c), thus reducing energy consumption. The energy consumption continues to decrease above R_{opt} until 10.24 s, but the degree of the decrease is low. As R reaches 10.24 s, the energy consumption tends to become constant. However, as we can see from Fig. 8, $T_{discover}$ increases severely above R_{opt} up to 10.24s. From these results, it can be stated that the proposed method does not only minimize $T_{discover}$, but also allows low energy consumption at nearly the lowest level possible.

V. CONCLUSION

BLE is one of the most promising technologies for use in the IoT because of its simplicity, low energy consumption, low cost, and long lifetime. As IoT is expected to become pervasive in everyday life, it is also expected that numerous BLE-enabled IoT applications and services will be developed in various areas. In such IoT services, scanners will be required to discover all surrounding advertisers in a short time. However, the requirement for fast discovery is very difficult to achieve when the number of advertisers is very large in a small area. As the number of advertisers increases, the signal density also increases, and the collisions among signals become more numerous. As the number of collisions that occur increases, so does the discovery delay. In this paper, we have proposed an analytical model to characterize the complex relationships between the discovery time and various BLE operational parameters such as the advertisement interval, scan interval, the number of advertisers, and so on. With the analytical model, we have also proposed a very effective method to determine the optimal advertisement interval to minimize the discovery time. With the proposed method, it has been shown that the discovery time can be reduced significantly by more than a factor of five compared with the case where default parameters are used.

As mentioned previously, many IoT services with large number of BLE devices are expected to emerge in the near future, involving many advertisers within a relatively small area such as a logistics warehouse, sports arena, subway, shopping mall, or other crowded places. The fast discovery of advertisers in crowded environments is essential to support such BLE-enabled services successfully. In addition to the fast discovery of advertisers, reducing the energy consumption of each advertiser should also be considered for providing the services. We expect that the proposed method can be effectively utilized as a criterion for the design and the provision of future BLE-enabled IoT services.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] R. Want, B. N. Schilit, and S. Jenson, "Enabling the Internet of Things," *Computer*, vol. 48, no. 1, pp. 28–35, Jan. 2015.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] *Bluetooth Core Specification Version 4.2*, Bluetooth SIG, Dec. 2014.

[5] *Bluetooth Core Specification Version 5.0*, Bluetooth SIG, Dec. 2016.

[6] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is Bluetooth low energy? Comparative measurements with ZigBee/802.15.4," in *Proc. IEEE WCNCW*, Paris, France, Apr. 2012, pp. 232–237.

[7] P. D. Martin, M. Rushanan, T. Tantillo, C. U. Lehmann, and A. D. Rubin, "Applications of secure location sensing in healthcare," in *Proc. ACM BCB*, Seattle, WA, USA, Oct. 2016, pp. 58–67.

[8] H.-W. Kang, C.-M. Kim, and S.-J. Koh, "ISO/IEEE 11073-based healthcare services over IoT platform using 6LoWPAN and BLE: Architecture and experimentation," in *Proc. NaNA*, Hakodate, Japan, Jul. 2016, pp. 313–318.

[9] G. Shan, B.-H. Park, S.-H. Nam, B. Kim, B.-H. Roh, and Y.-B. Ko, "A 3-dimensional triangulation scheme to improve the accuracy of indoor localization for IoT services," in *Proc. PACRIM*, Victoria, BC, Canada, Aug. 2015, pp. 359–363.

[10] T. A. Johnson and P. Seeling, "Localization using Bluetooth device names," in *Proc. ACM MobiHoc*, Hilton Head, SC, USA, Jun. 2012, pp. 247–248.

[11] R. A. Ramlee, D. H. Z. Tang, and M. M. Ismail, "Smart home system for disabled people via wireless Bluetooth," in *Proc. ICSET*, Bandung, Indonesia, Sep. 2012, pp. 1–4.

[12] M. Collotta and G. Pau, "Bluetooth for Internet of Things: A fuzzy approach to improve power management in smart homes," *Comput. Elect. Eng.*, vol. 44, pp. 137–152, May 2015.

[13] N. Newman, "Apple iBeacon technology briefing," *J. Direct, Data Digit. Marketing Practice*, vol. 15, no. 3, pp. 222–225, Jan. 2014.

[14] J. Decuir, "Introducing Bluetooth Smart: Part II: Applications and updates," *IEEE Consum. Electron. Mag.*, vol. 3, no. 2, pp. 25–29, Apr. 2014.

[15] J. Liu, C. Chen, Y. Ma, and Y. Xu, "Adaptive device discovery in Bluetooth low energy networks," in *Proc. IEEE VTC Spring*, Dresden, Germany, Jun. 2013, pp. 1–5.

[16] J. Kim and K. Han, "Backoff scheme for crowded Bluetooth low energy networks," *IET Commun.*, vol. 11, no. 4, pp. 548–557, Mar. 2017.

[17] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, Aug. 2012.

[18] G. Shan, B.-U. Lee, S.-H. Shin, and B.-H. Roh, "Design and implementation of simulator for analysis of BLE broadcast signal collision," in *Proc. ICOIN*, Da Nang, Vietnam, Jan. 2017, pp. 448–452.

[19] A. Balogh and S. Imre, "Simulation and analysis of concurrent BLE link layer state machines running within the same physical device," in *Proc. SoftCOM*, Split, Croatia, Nov. 2017, pp. 1–8.

[20] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in Bluetooth low energy networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1439–1441, Sep. 2012.

[21] J. Liu, C. Chen, and Y. Ma, "Modeling and performance analysis of device discovery in Bluetooth low energy networks," in *Proc. IEEE GLOBECOM*, Anaheim, CA, USA, Dec. 2012, pp. 1538–1543.

[22] K. Cho *et al.*, "Analysis of latency performance of Bluetooth Low Energy (BLE) networks," *Sensors*, vol. 15, no. 1, pp. 59–78, 2015.

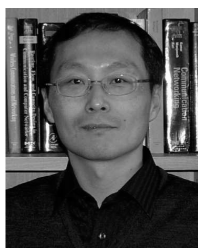
[23] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance analysis of neighbor discovery process in Bluetooth low-energy networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1865–1871, Feb. 2017.

[24] R. Schrader, T. Ax, C. Röhrig, and C. Fühner, "Advertising power consumption of Bluetooth low energy systems," in *Proc. IDAACS-SWS*, Offenburg, Germany, Sep. 2016, pp. 62–68.

[25] J. Liu, C. Chen, Y. Ma, and Y. Xu, "Energy analysis of device discovery for Bluetooth low energy," in *Proc. IEEE VTC Fall*, Las Vegas, NV, USA, Sep. 2013, pp. 1–5.

[26] Á. Hernández-Solana, D. Perez-Diaz-de-Cerio, A. Valdovinos, and J. L. Valenzuela, "Proposal and evaluation of BLE discovery process based on new features of Bluetooth 5.0," *Sensors*, vol. 17, no. 9, p. 1988, Jul. 2017.

GAOYANG SHAN was born in Henan, China, in 1990. He received the B.S. degree in electronics engineering from Kyonggi University, Suwon, South Korea, in 2014, and the M.S. degree in computer engineering from Ajou University, Suwon, in 2016, where he is currently pursuing the Ph.D. degree in computer engineering with the Graduate School. His research interests include mobile wireless networks and Internet of Things.



BYEONG-HEE ROH (S'95–A'99–M'04–SM'17) received the B.S. degree in electronics engineering from Hanyang University, Seoul, South Korea, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1989 and 1998, respectively. From 1989 to 1994, he was with the Telecommunication Networks Laboratory, Korea Telecom, as a Researcher. From 1998 to 2000, he was with with Samsung Electronics Co., Ltd., South Korea, as a Senior Engineer. In 2005, he was a Visiting Associate Professor with the Department of Computer Science, State University of New York, Stony Brook, NY, USA. In 2014, he was an Adjunct Researcher with the Agency for Defense Development, South Korea. Since 2000, he has been with the Department of Software and Computer Engineering and the Department of Software and Computer Engineering, Graduate School, Ajou University, Suwon, South Korea, where he is currently a Professor. His research interests include mobile multimedia networking, future Internet networks, Internet of Things platform and services, network security and cyber warfare, and military communications.