

Article

Efficient Advertiser Discovery in Bluetooth Low Energy Devices

Seung Whan Song ¹, Youn Sang Lee ¹, Fatima Imdad ², Muhammad Tabish Niaz ³  and Hyung Seok Kim ^{2,*}

¹ SOOSAN ENS Co. Ltd, Hyundai Venture Ville 3F 304, 10, Bamgogae-ro 1-gil, Gangnam-gu, Seoul 06349, Korea; songplc01@soosan.co.kr (S.W.S.); leeys01@soosan.co.kr (Y.S.L.)

² Department of Information and Communication Engineering, Sejong University, 209 Neungdong-Ro, Gwangjin-gu, Seoul 05004, Korea; fatimaimdad@sju.ac.kr

³ Department of Smart Device Engineering, Sejong University, 209 Neungdong-Ro, Gwangjin-gu, Seoul 05004, Korea; mtabishniaz@sejong.ac.kr

* Correspondence: hyungkim@sejong.ac.kr; Tel.: +82-2-3408-3696

Received: 15 April 2019; Accepted: 30 April 2019; Published: 6 May 2019



Abstract: Bluetooth Low Energy (BLE) has become ubiquitous in the majority of mobile devices that connect wirelessly. With the increase in the number of devices, the probability of congestion also increases in a network. Data channels of the BLE use frequency hopping, but it is not available for advertising channels. The capability of the BLE for providing a wide range of parameters settings ensures the impressive potential for BLE devices to customize their discovery latency. But communication before connection setup is not synchronous and both the scanning devices and the advertising devices are unaware of the timing parameters of each other. This can lead to inefficient advertiser device discovery. To resolve this issue, an algorithm is proposed to reduce the average latency per advertiser experienced due to the increase in the number of BLE devices in a vicinity. It is observed that the average latency has shown improvement in the range of 35% to 55%, depending on different simulated scenarios. Due to this improvement the overall energy consumption is also reduced.

Keywords: Bluetooth Low Energy; Bluetooth; Advertiser Discovery; Internet of Things; the neighbor discovery process; minimum discovery time

1. Introduction

As wireless communication technology is entering its 5th generation (5G) [1,2], there has been an increase in the number of Internet of Things (IoT) applications. With this increase in IoT applications, the number of physical devices is also continuously growing [3]. To enable these IoT applications, there have been a lot of studies mainly covering two aspects of communication technologies and networking technologies. Some known examples of communication technologies for IoT applications include Bluetooth, Bluetooth Low Energy (BLE), ZigBee, and radio-frequency identifiers (RFIDs). On the other hand, some examples for networking technologies include IPv6 over Low Power Wireless Personal Area Network (6LowPAN), low power and lossy networks (RPL), and others [4,5]. As the above-mentioned technologies have their own protocols and distinct features, their use is based on the type of application.

BLE, commercially known as Bluetooth Smart, is an evolving low-power wireless technology for short-range communication. It has been developed by Special Interest Group (SIG). BLE operates on the Industrial Scientific Medical (ISM) band of 2.4 GHz [6]. A newer version of BLE [7] aims to make it even better for IoT because it provides low power internet connectivity. This is achieved through the

use of IPv6/6LoWPAN. BLE is well suited for systems running on small batteries, as the key feature of this technology is lower power consumption [8]. Hence the devices running on BLE can use a coin-cell as their power source and run for years. Its characteristics, like low power consumption and low cost, make it an ideal choice for power sensitive sensor devices. BLE can be incorporated in a wide range of applications such as sensor-enabled environmental and healthcare monitoring [9–12], smart home and automation [13], indoor localization [14,15], tagging, and proximity-based services such as near-field communication (NFC) or RFID [16], and others [17].

In BLE applications, multiple BLE slave devices, known as advertisers, can connect to a master BLE device, known as a scanner. The topology is usually a star topology piconet. To associate a slave device to a master device, a neighbor discovery process (NDP) is usually initiated. The NDP is defined in both BLE 4.2 and 5.1. The process of NDP is very simple, it sends advertisement packets over three advertisement channels periodically to its scanner for a certain BLE-enabled service. Because BLE uses an ISM band, it can be freely operated in crowded environments to establish a network [18]. As the number of BLE devices to be connected to a specific network increases, the number of collisions will also increase. This, in turn, reduces the battery life and increases the time for discovery of individual advertisers [19]. Due to this factor, the fast discovery of advertisers is a challenging problem in BLE.

The BLE specifications allow a wide range of parameters that can be adjusted according to the mode and scenario that it is deployed for. Amongst these parameters, there are also a few parameters which can be adjusted for the BLE NDP process [8]. Utilizing this freedom granted in the BLE specifications, there have been many studies which have analyzed the discovery time performances using simulators [20,21], and analytical models [18,19,22–25]. In [18,22–24], the authors have presented an analytical model for the discovery time taken by a scanner to discover a targeted advertiser with a probability model. The authors of [25], used power consumption models based on [26] for the discovery process. The authors in [25] analyzed the trade-off between discovery time and power consumption. A back-off strategy of BLE devices is proposed in [19], it reduces the collision probability and enhances the discovery latency performance during the NDP stage. These studies usually only focus on the advertiser and have tried to reduce the power consumption and discovery time at the advertiser only. It is described in [25], that the optimal setting of the BLE devices to minimize the discovery time is essential for the development and smooth operations of BLE-enabled IoT infrastructure.

Lowering the energy consumption of BLE-enabled IoT applications is also one of an important design requirement. With multiple operation modes available in BLE-enabled devices, the standard itself is very energy efficient. But due to certain network scenarios the energy efficiency sometimes goes down. The energy consumption parameters directly affect the lifetime of the BLE-enabled IoT devices. Hence, it is a very important topic to discuss when dealing with BLE. A power consumption model was proposed in [26], which analyzed the energy consumption during the NDP phase. A performance model was proposed in [25] which evaluated the trade-off between the BLE discovery time and energy consumption. The authors of [27] proposed a study on the power consumption model of BLE NDP process based on different advertising parameters of BLE. A back off strategy for BLE devices based on the number of collision probability was analyzed in [19] to enhance the BLE NDP performance.

In this paper, an algorithm is proposed in which BLE scan interval time and scan window times are adjusted for optimal discovery time of advertisers. The BLE specification allows some protocol data unit (PDUs) to send information. The advertiser can use this feature and send information to the scanner regarding its current parameters. Based on the information the scanner adjusts the scan interval and the scan window times so that the overall advertiser discovery time is reduced and hence energy is saved. The proposed algorithm is different from the existing work in the manner that it utilizes the freedom of setting the parameters range according to the current network situation. It also uses the option to send user data in PDU from advertiser to scanner. Using this option, the algorithm presented in this paper utilizes the runtime information exchanged between the advertiser and scanner, to adjust the scan interval and scan window time. The energy consumption of the proposed method is also analyzed. To the best of the authors' knowledge, this work has not been done before.

The rest of the paper is organized as follows: the background and the NDP process in BLE are described in Section 2. In Section 3, the proposed solution is described. The simulation results are presented in Section 4, while Section 5 will conclude the paper.

2. Background

2.1. BLE Neighbor Discovery Process

BLE operates in the ISM 2.4 GHz (2402 MHz ~ 2480 MHz band). The total number of available channels is 40, out of which 37 are used for data communication after the setup of the connection. The remaining three channels (37, 38, and 39) are used for advertising purposes. The Link Layer of BLE is expressed in terms of a state machine, all five states are standby, advertising, scanning, initiating, and connection state. When a device is not sending or receiving any packets it is said to be in a standby state. When the BLE device is in advertising mode it is called an advertiser. The advertising channels are used to send advertising information in the form of advertising events. The time between two consecutive advertising events is denoted by T_{Event} . This value is a combination of two parameters; the advertising interval $T_{AdvInterval}$ and a random delay $T_{AdvDelay}$. A scanner is the device which is operated in scanning mode. The purpose of scanning mode is to listen for advertising packets from surrounding advertisers. It also has the role of inquiring additional information from an advertiser or totally ignoring it. If a device is in initiating state, it means it is listening for advertiser's packets and responds to them so that a connection is made with the other device. In the case of NDP, the BLE device working under initiator or scanner will have identical operation [25].

As per the BLE specifications, there are typically two types of advertising events: undirected and directed. If the connecting device is already known to the master device, a directed advertising event is considered. But in the case when the connecting device is new or unknown to the master device then it is said to be an undirected advertising event [18]. The undirected advertising event is further divided into three types: scannable, connectable, and non-connectable events [6,7]. In the case of scannable and connectable events, the scanning or initiating device may send out additional information or request for connection establishment to advertisers. In the case of non-connectable undirected advertising events, no request or response from scanning, or initiating device is transmitted to the advertising device.

BLE allows a wide range of setting parameters for advertising and scanning events, as shown in Figures 1 and 2. The $T_{AdvInterval}$ is the time between two consecutive advertising events minus the time of $T_{AdvDelay}$. $T_{AdvInterval}$ can be of any value in between 20 ms to 10.24 s. The value of $T_{AdvInterval}$ should always be a multiple of 0.625 ms. However, there is a restriction on the event types, scannable undirected and non-connectable undirected, the value of $T_{AdvInterval}$ cannot be less than 100 ms. $T_{AdvDelay}$ is a pseudo-random value that ranges between 0 to 10 ms.

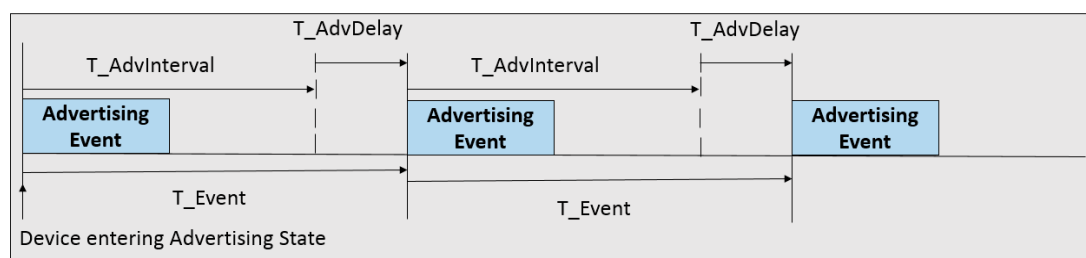


Figure 1. Timing specifications for an advertising event.

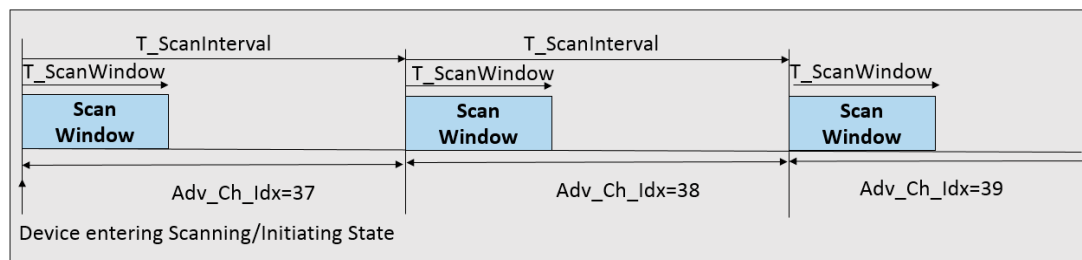


Figure 2. Timing specification for scanning event.

There are four advertising event types. Each one corresponds to an advertising PDU type. Some allow a response from the scanner, some from the initiator, and some do not allow the advertiser to scan its channels to receive data. The device is said to be a scanner when its link layer is in the scanning state. Like the advertiser, it listens to each advertising channel index. The duration for which the scanner listens to the channel is called a scan window denoted by $T_{ScanWindow}$. The scan interval is the time interval between two scan windows, it is denoted by $T_{ScanInterval}$. Both the values of $T_{ScanWindow}$ and $T_{ScanInterval}$ should be less than or equal to 10.24 s. If both values are equal, then the scanning mode becomes continuous. The continuous scanning mode drains the battery; hence it is discouraged but as per specifications it can be implemented. The initiating state works in the same way as scanning state, the only difference is that devices in both states respond to distinct advertising packets and a connection is established after the interaction of advertiser with initiator.

At this point let's discuss the issue of interference with some other wireless technologies using the same ISM band (WiFi). According to different studies done on the interference between BLE and WiFi [28] it is seen that these two technology can coexist without any interference issue. The BLE advertising channel are located at 2402 MHz, 2426 MHz, and 2480 MHz. These channels are carefully placed in those specific frequencies so that they don't interfere with the 3 main WiFi channels in 2.4 GHz band. For example, the WiFi channel 6 is centered at 2437 MHz and is 22 MHz wide, so ranging from 2427 MHz to 2448 MHz. For this particular channel, the data utilizes only 20 MHz. And the other 2 MHz are used for guard band so that there is enough attenuation along the edge channel. As channels 1, 6, and 11 are the most commonly used channels, they do not overlap and there is not data transmitted on the channel 2426 MHz. The reason for this is that WiFi will see BLE as a narrow band interference and only in the cases that the BLE interference is not more than 10 dBm below the WiFi signal can occasionally cause interference. Usually WiFi transmit in 20 dBm and BLE transmit at 0 dBm. With these values theoretically there will be an interference. But the channel hopping mechanism in BLE standard helps avoid any interference. Thus, both WiFi and BLE can safely coexist in the same area.

2.2. BLE Neighbor Discovery Process

As BLE NDP has adjustable parameters, the choice of the parameter settings affects the discovery latency performance as discussed in the researches [18], and [24]. The authors in [18] introduced an analytical model based on pure ALOHA for continuous scanning mode. The research showed that discovery latency is directly proportional to the number of BLE devices. In some cases, it even increases exponentially. The authors in [24] proposed an analytical model to calculate the discovery probability. It also investigated how the settings of BLE parameters influenced the overall system performance. All these researches are a little old as they considered BLE version 4.0. Up till now the major consumer market device are using BLE 4.2 [6] and recently BLE 5.0 [7] was introduced. The operations of NDP in version 4.2 and 5.0 are almost the same except the new version allows greater freedom to adjust a few parameters of BLE NDP. Also, there are two types of NDPs defined in version 5.0 of BLE: basic and advanced. The basic version is almost the same as of BLE 4.2 except the allowable range of BLE parameters.

3. Proposed Efficient BLE Advertiser Discovery Solution

In this section, the problem statement is formulated, and the proposed solution is explained. For this section, we will assume that there is one scanner and there are N advertisers. All of the N advertisers lie in the discovery range of the scanner. We also assume that only undirected advertising events from advertisers are sent. As mentioned in the previous section, Section 2, there is no significant difference in advertising and scanner events in BLE NDP. For the proposed solution it is also assumed that there is fair contention for the discovery and each advertiser is independent of other.

3.1. Problem Statement

Discovering of the advertiser by a scanner and the initiator is of paramount importance in BLE communication. Before the setup of the connection, the initiator needs to discover the advertising device. Similarly, given the condition that there is no connection required, if two BLE devices want to communicate, it is only possible if the advertising device is discovered by the scanner in the appropriate time and the information is sent using the payload of advertising PDUs.

There are many devices that have BLE technology enabled for them and the number is increasing day by day. In a denser environment, this discovery can take more time. Especially if the devices are not familiar with each other, in other words, not previously discovered by each other. BLE devices do not only suffer due to more devices of BLE in nature but also other technologies can influence the time consumed to be discovered and connection setup time. This is because BLE shares the ISM band with many other technologies.

BLE has dealt with this issue by incorporating frequency hopping schemes on the channels. But unfortunately, this is only applicable to the data channels [5]. This leaves the 3 advertising channels more prone to interference and contention. Carrier Sense Multiple Access (CSMA) is also not available in BLE.

In a dense environment, there can be a high probability of broadcasting conflicts. The performance of BLE deteriorates if the number of devices in the vicinity is increased. But what is of more concern is that if the parameters like the timing of PDUs are not set appropriately there is more probability of service degradation in terms of latency.

The scanning device may scan continuously, but this is highly discouraged as it consumes a lot of energy and cuts down the battery time of the host device. And even then, the scanning device is not familiar with the conditions or the settings at the advertising end. This can result in completely out of synchronization communication and inefficient discovery in terms of latency and energy.

When the number of devices increases then the issue becomes more severe and average discovery time increases with a great extent [29]. Considering a scenario where there are numerous BLE devices waiting to be discovered, there is no way they can be efficiently discovered unless the network conditions are considered and then worked upon. Considering applications like iBeacon, where the communication actions occur only through advertising channels and packets, the traffic on advertising channels becomes dense.

Considering all these issues, this paper proposes an algorithm as a solution. The main task of the algorithm is to dynamically adjust the window size of the scanner so that the advertiser discovery time is reduced thus increasing the efficiency of BLE network.

3.2. Proposed Solution

An algorithm is proposed to make the scanner or initiator aware of the conditions of the network by sending very small information to them. By exploiting this information, parameters like scan window size and scan interval are adjusted accordingly. The information is sent by encapsulating it into the advertising PDUs. There are four available PDUs, namely *DV_DIRECT_IND*, *ADV_NONCONN_IND*, *ADV_IND*, and *ADV_SCAN_IND*.

Some of the PDUs have the capability of sending data even without connection establishment. Only *ADV_DIRECT_IND* PDU type cannot contain data in it. On the other hand, the rest of the three PDUs can encapsulate data in their packets. PDU *ADV_NONCONN_IND* does not allow the advertiser to scan the advertising channels after the PDU is sent. If the advertising PDU type is a non-connectable undirected event, then it would not receive *SCAN_REQ* and *CONNECT_REQ*. It only sends advertising PDUs continuously on the advertising channel indices.

This leaves us with two advertising PDUs in which data can be encapsulated and a response can be received. We use *ADV_IND* and *ADV_SCAN_IND* PDU types to accommodate a variable from each advertiser and deliver it to scanner and initiator.

When *ADV_SCAN_IND* is sent to a scanner, the scanner responds with *SCAN_REQ*. As a response to *SCAN_REQ*, the advertiser sends *SCAN_RSP* as shown in Figure 3. When the *ADV_IND* is sent to an initiator, the initiator responds with *CONNECT_REQ*. After the successful exchange of these packets, the connection is established. The advertiser and initiator both go into their connection state, as shown in the Figure 4.

We define the latency of advertiser to be discovered by the scanner, as the time when advertiser went into advertising state to the time when *SCAN_REQ* is received by the advertiser. We also define the latency of advertiser to be discovered by the initiator as, the time when the advertiser went into advertising state to the time when *CONNECT_REQ* is received by the advertiser.

As per BLE specifications, there is an inter-frame space (T_{IFS}) that is defined as the time interval between two consecutive packets on the same channel index. T_{IFS} is equal to 150 μ s.

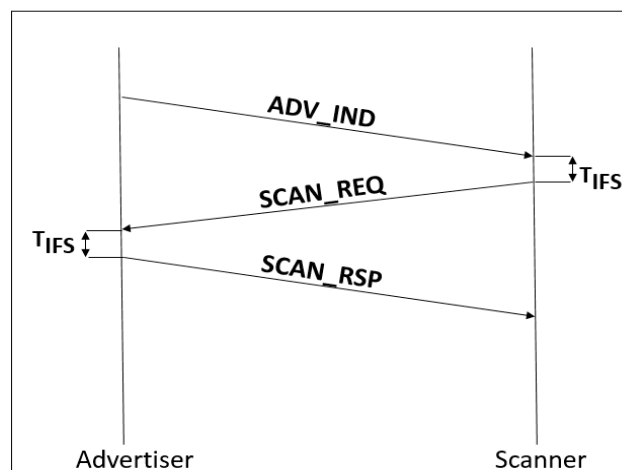


Figure 3. Advertiser discovery by the scanner.

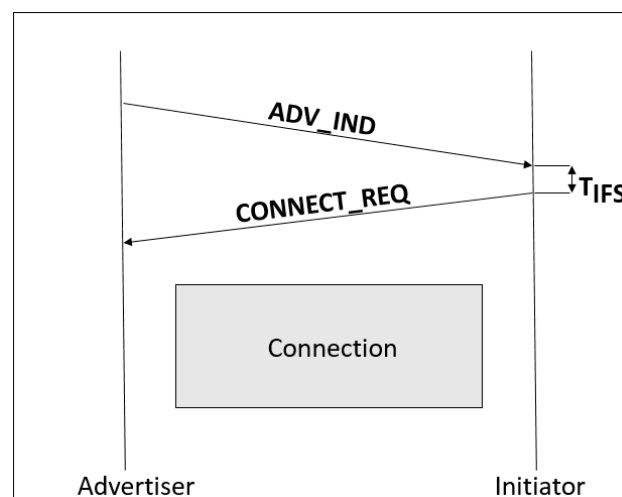


Figure 4. Advertiser discovery by Initiator.

3.3. Algorithm

The proposed Algorithm 1, is formulated around an indicator sent by the advertiser. The indicator is denoted as Adv_C . After the advertiser enters its advertising state, initially the value of Adv_C is set to zero and it is incremented after every passing advertising interval. This value is set to zero when a response is received from either a scanner or an initiator. The value of Adv_C is sent in the payload of ADV_IND and ADV_SCAN_IND PDUs. These PDUs are received at the scanner or initiator. The scanner accesses that, after how many tries of sending PDUs, the advertiser was successfully discovered by the scanner. This systematic approach implies that if the value of Adv_C is very high then it took several attempts to reach the scanner or initiator. In turn, the scanner (or initiator) readjusts its scan window ($T_{ScanWindow}$) and scan interval ($T_{ScanInterval}$) accordingly.

A threshold is set for the maximum value of Adv_C ; this is termed as Adv_{C_MAX} . The changes in the value of $T_{ScanWindow}$ and $T_{ScanInterval}$ are based upon the ratio of $T_{ScanInterval}$ to $T_{ScanWindow}$ i.e., δ .

$$\frac{T_{ScanInterval}}{T_{ScanWindow}} = \delta \quad (1)$$

If the value of Adv_C is equal to or greater than Adv_{C_MAX} then the value of $T_{ScanInterval}$ is decreased with a factor of δ . If the value of Adv_C is less than Adv_{C_MAX} then $T_{ScanWindow}$ is decreased with a factor of δ and $T_{ScanInterval}$ is increased by a factor of δ .

The value of Adv_C is saved in the scanner and becomes Adv_{CP} . When this value is received for a new advertiser, Adv_C and Adv_{CP} are compared, if the value for new advertiser is greater than one previously stored. Then the $T_{ScanInterval}$ is again decreased with a factor of δ and $T_{ScanWindow}$ is increased with a factor of δ .

Lower thresholds are set for both $T_{ScanInterval}$ and $T_{ScanWindow}$ and if they are decreased beyond these thresholds then the values are set to T_{SI} and T_{SW} respectively. In the case of new value scan window being greater than the scan interval, scan interval gets the value of scan window. And the scan window is adjusted with respect to T_{Buffer} . The upper limit of both $T_{ScanInterval}$ and $T_{ScanWindow}$ is set to be 10.24 s as per the BLE 4.2 specifications.

Another condition considered is that the value of delta is never 1, because this means that the scanner is scanning continuously. Continuous scanning is allowed as per BLE specification but it costs energy, this algorithm avoids continuous scanning by never letting the value of δ to be 1.

Algorithm 1. Proposed Algorithm

Function (ADV_IND , ADV_SCAN_IND , Adv_C , $T_{ScanInterval}$, $T_{ScanWindow}$, Adv_{CP}).

Input: Payload from advertiser; ADV_IND or ADV_SCAN_IND , Indicator from advertiser; Adv_C , value of scan interval; $T_{ScanInterval}$, value of scan window; $T_{ScanWindow}$, previous value of indicator Adv_{CP} .

Output: refined value of scan interval; $T_{ScanInterval}$ and refined value of scan window; $T_{ScanWindow}$.

1 **get** Adv_C from PDUs.

$\delta \leftarrow \frac{T_{ScanInterval}}{T_{ScanWindow}}$

3 **if** ($Adv_C \geq Adv_{C_MAX}$) **then**

4 $T_{ScanInterval} \leftarrow T_{ScanInterval}$ is decreased by a factor of δ

5 **else if** ($Adv_C > Adv_{CP}$) **then**

6 $T_{ScanInterval} \leftarrow T_{ScanInterval}$ is decreased by a factor of δ

7 $T_{ScanWindow} \leftarrow T_{ScanWindow}$ is increased by a factor of δ

8 **else if** ($Adv_C < Adv_{C_MAX}$) **then**

9 $T_{ScanWindow} \leftarrow T_{ScanWindow}$ is decreased by a factor of δ

10 $T_{ScanInterval} \leftarrow T_{ScanInterval}$ is increased by a factor of δ

11 **else if** ($T_{ScanWindow} > T_{ScanInterval}$) **then**

12 $T_{ScanInterval} \leftarrow T_{ScanWindow}$

13 $T_{ScanWindow} \leftarrow T_{ScanWindow} - T_{Buffer}$

14 **return** $T_{ScanInterval}$, $T_{ScanWindow}$

15 **end**

3.4. Parameter Calculations

The total size of the BLE packet is $1 + 4 + (2 \sim 257) + 3$ octets (Preamble + Access Address + PDU + CRC), this is shown in Figure 5. The overall packet structure for both advertising and data packets is the same, with the difference lying in the PDU field. The length of advertising PDU depends upon the length field in 2 octet header. Considering PDUs *ADV_IND* and *ADV_SCAN_IND*, the maximum size of these packets will be 47 octets ($1 + 4 + 3 + 2 + 6 + 31$). As per specifications, the maximum packet size of *SCAN_REQ* is 22 octets ($1 + 4 + 3 + 2 + 6 + 6$). Similarly, the packet size of *CONNECT_REQ* is calculated to be 44 octets ($1 + 4 + 3 + 2 + 6 + 6 + 22$). It is to be noted that the packet *CONNECT_REQ* is most prone to error due to the presence of crucial data in LLData field. If it gets corrupt then the probability of discovery is reduced thus adding to the latency.

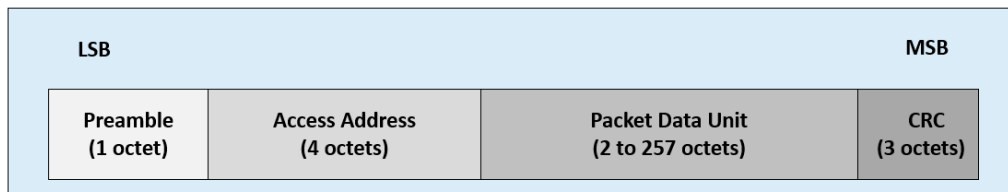


Figure 5. Bluetooth Low Energy (BLE) packet structure.

The transmission time of *ADV_IND* and *ADV_SCAN_IND* is denoted by T_a . The transmission time of *SCAN_REQ* and *CONNECT_REQ* are denoted by T_s and T_c respectively. Because the transmission rate for BLE is 1 Mbps the transmission times are calculated as;

$$T_a = \frac{47 \times 8}{1 \text{ Mbps}} = 376 \mu\text{s} \quad (2)$$

$$T_s = \frac{22 \times 8}{1 \text{ Mbps}} = 176 \mu\text{s} \quad (3)$$

$$T_c = \frac{44 \times 8}{1 \text{ Mbps}} = 352 \mu\text{s} \quad (4)$$

Keeping in view the value of inter-frame space, the minimum time required by the advertiser to be discovered T_d is:

$$T_d = T_a + T_s + T_{IFS} + \text{other delays} \quad (5)$$

Similarly, the latency of advertiser to be discovered by the initiator can be stated as:

$$T_a + T_c + T_{IFS} + \text{other delays} \quad (6)$$

These calculations are used in the simulation to simulate a realistic scenario of BLE network, which is also in accordance with the BLE specifications.

3.5. Energy Consumption

The calculation of the energy consumption is depended on multiple parameters. Since the scope of this paper is limited to the advertiser discovery we will only deal with the calculation of this event. The advertiser discovery is usually denoted by T_d . To carry on with the calculation we are assuming that the operational voltage of a particular advertiser is constant throughout the discovery event. The energy consumption is defined by the elapsed time (ms) multiplied by the electric current (mA). The research published by the authors in [25,26] are considered to be the most accurate work done for Bluetooth advertiser discovery energy consumption. Based on the model proposed in these researches, the energy consumption of an advertiser in a single advertisement period is E_a .

Using the value of E_a , the expected energy consumption of an advertiser during the interval T_d is given as:

$$E_{OneAdv} = \frac{T_d}{T_{AdvInterval} + E[T_d]} \times E_a \quad (7)$$

The total expected energy consumption for all N advertiser is computed using the following equation:

$$E_{Total} = E_{OneAdv} \times N \quad (8)$$

4. Simulation and Results

To test the results of the proposed algorithm, a BLE simulating program was developed. The simulation program was developed using C language and Matlab. The choice of using C language was to avoid additional software delays during the simulation process. The simulation results were imported to Matlab for further analysis. The simulation complies with all the requirements of the BLE 4.2 specifications document. In order to highlight the improvement made by the proposed algorithm, the simulation is compared with scenarios of standard BLE. It is assumed that the BLE-enabled IoT devices are based indoor and they are usually static. Since a small home office scenario is considered it is also assumed that all the BLE-enabled IoT device are within the operating range of the BLE standard having good RSSI levels.

The number of advertisers used for the simulation are 10, 30, 50, 70, 90, and 100. Simulation for each distinct scenario has been run 100 times and the results have been averaged out. This is done to achieve more accurate results. As a vast number of devices are being considered in the scenario, there is a big variation in results as the number of devices increase. The results have been plotted on a logarithmic scale.

The simulation was run separately for four scenarios. First, the scenario of one scanner with a multiple number of advertisers is considered. Here the advertising devices enter their advertising state randomly and wait to be discovered by the scanner. Similarly, the scenario of one initiating device with a multiple number of advertisers was considered. Here the advertising devices started the transmission of their packets randomly and waited to be discovered by the initiator.

The time taken for an advertiser to be discovered by a scanner and an initiator is different. This difference is due to the different natures of exchanged packets. The *SCAN_REQ* is sent to the advertiser as a response to its PDUs. While *CONNECT_REQ* is sent by the initiator. Lengths of both these packets are different. Moreover *CONNECT_REQ* contains data, this data if not received correctly can lead to the case of non-discovery. Simulation parameters are stated in Table 1.

Table 1. Simulation Parameters.

Parameters	Value
Advertising Interval ($T_{AdvInterval}$)	200 ms
Base Scan Interval ($T_{ScanInterval}$)	5 s
Base Scan Window ($T_{ScanWindow}$)	1 s
Inter Frame Space (T_{IFS})	150 μ s
Number of Advertisers	10 ~ 100
$E(T_d)$	5 ms
$T \times \text{Power}$	4 dBm

The latency of a number of advertisers to be discovered by the scanner and initiator are shown in different graphs. For the discussion of the results, the average delay means the average time in seconds it took for one advertiser to be discovered by the scanner or initiator. The total delay means the total time it took for all the advertisers to be discovered. This value is the aggregated time of all the advertisers from the moment they individually entered the advertising state to the moment they received the response from scanner or initiator. Another parameter that is discussed in the results is the maximum time one advertiser had to wait to be discovered. All three of these parameters show the efficiency of our algorithm in comparison to an ordinary scenario with no algorithm applied.

The worst-case scenario was simulated for both the scanning device and the initiating device. In this simulated scenario, all the advertising devices start transmission at the same time. This increases the average waiting time per advertiser and the maximum waiting time drastically.

Firstly, we discuss the scenario where the scanning device is scanning multiple number of advertisers. As the number of devices increase, the total time taken by all the advertisers, from the time when they entered their advertising state till the time they receive SCAN_REQ from the scanner, is observed to be increasing. Figure 6 shows the total time taken by 10 to 100 advertising devices to be discovered by the scanning device. Here, the advertisers start transmission at a random time. This graph shows the comparison of total time taken without the algorithm application and with the algorithm application.

The total time it took for 10 devices was 9.2942 s in the ordinary scenario. It took 370.8515 s for 100 devices. These values have been decreased to 3.0969 s and 130.9671 s respectively after the application of the proposed algorithm. The total time taken by all the advertisers to be discovered by the scanner is decreased by 35.31% in the case of 100 devices.

In Figure 7, the graph shows the improvement in the average discovery time for 10 to 100 devices. The average discovery delay for 100 devices was decreased from 3.7 s to 1.3 s after the application of the proposed algorithm.

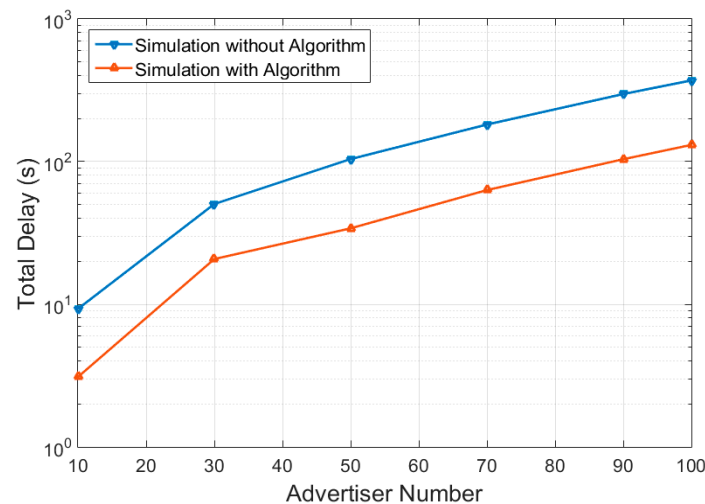


Figure 6. Total delay for advertiser discovery by the scanner.

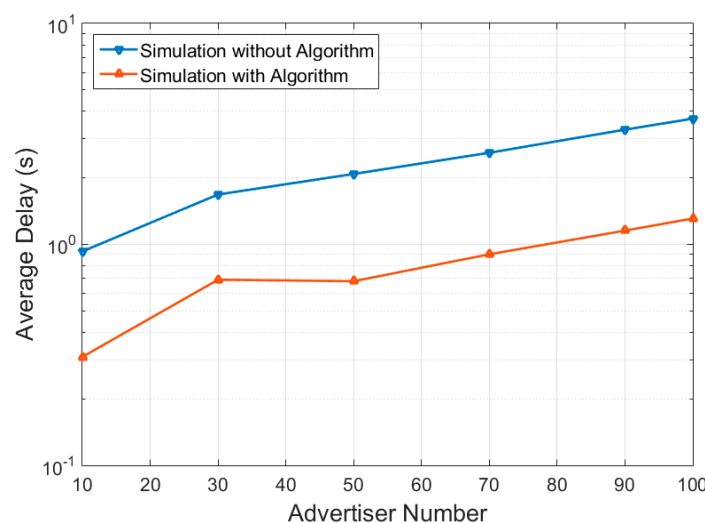


Figure 7. Average delay for advertiser discovery by the scanner.

The same scenario was also simulated for an initiator. The advertising devices randomly enter their advertising state and start the transmission of packets. The results of this scenario can be seen in Figures 8 and 9. Figure 8 shows the total time taken for all the advertisers to be discovered by the initiator. The value increases as the number of advertising devices are increased. On average the improvement in total delay is 34.5% for total time taken for all the advertisers to be discovered. As per the results of the average delay, from Figure 9, the average delay for an advertiser to be discovered has been decreased from 6 s to 2.103 s.

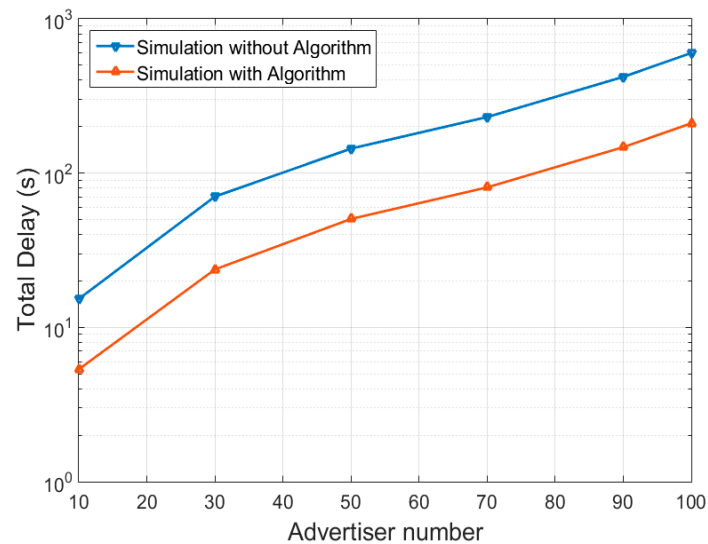


Figure 8. Total delay for advertiser discovery by the initiator.

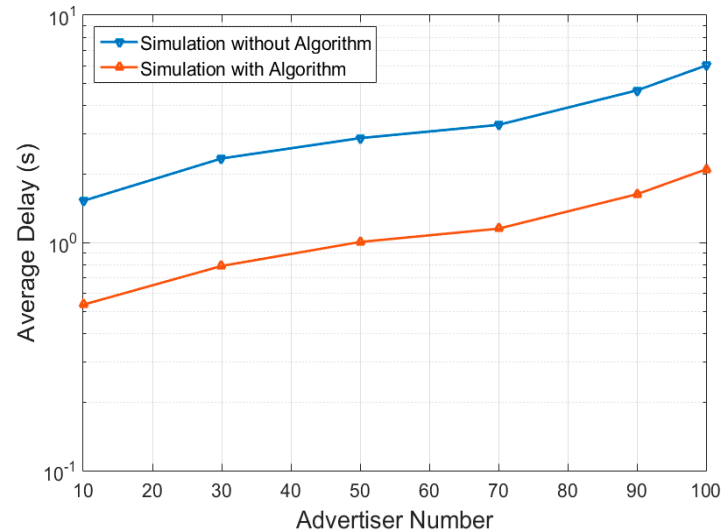


Figure 9. Average Delay for advertiser discovery by the initiator.

The worst-case scenario was also simulated considering that all the advertising devices start their transmission at one time. This showed a drastic increase in the average discovery delay. The maximum waiting time was also computed for these advertisers. In Figures 10–12 the total delay, the average delay and the maximum time an advertiser had to wait before being discovered are shown. Total time taken by 100 advertisers to be discovered by a scanner is improved by 55.9% when the algorithm was applied. The average discovery delay per advertiser was reduced from 4.106 s to 2.296 s. This result can be seen in Figure 11.

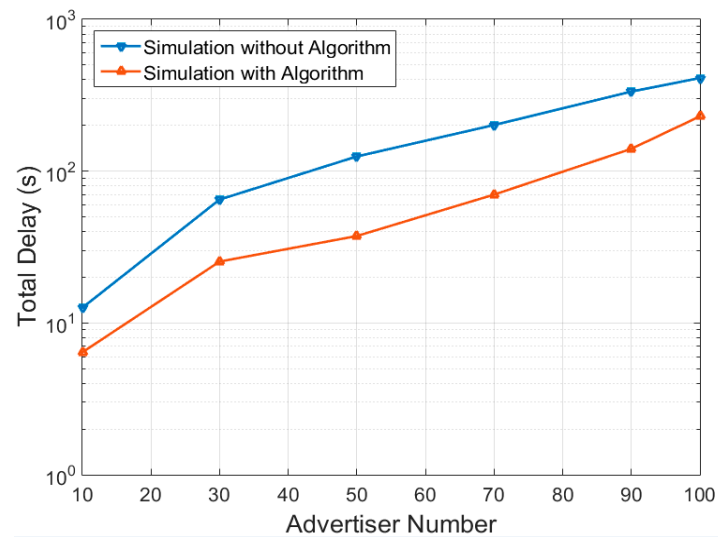


Figure 10. Total delay for advertiser discovery by the scanner in the worst-case scenario.

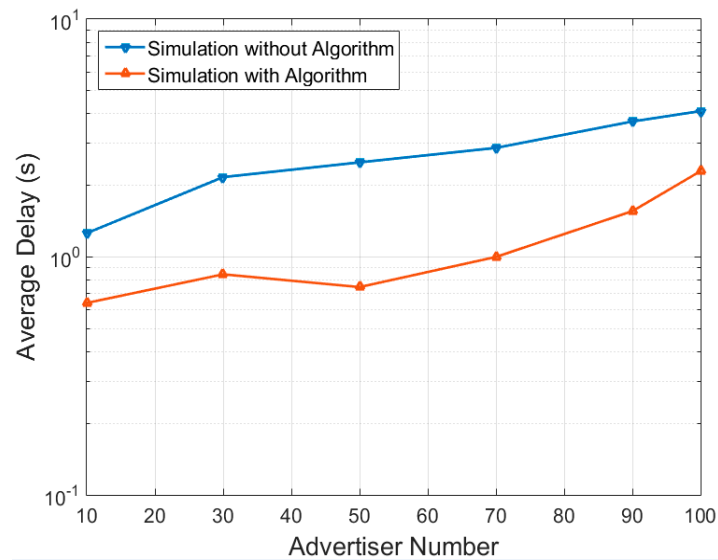


Figure 11. Average delay for advertiser discovery by the scanner in the worst case scenario.

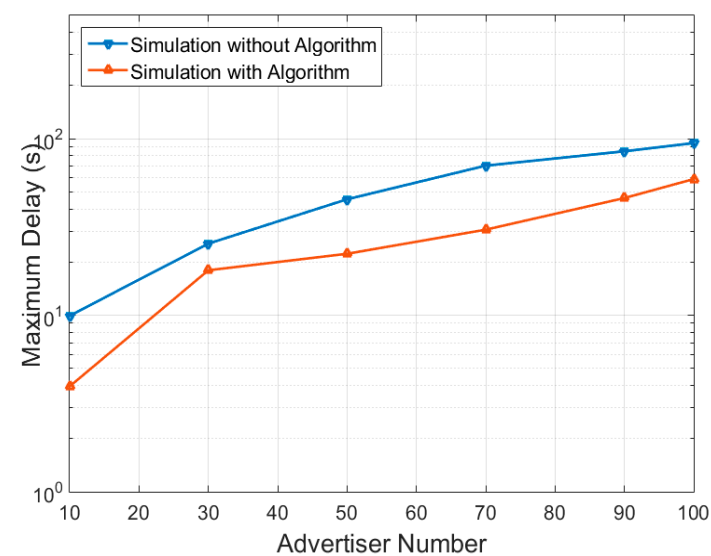


Figure 12. Maximum delay of one advertiser discovery by the scanner in the worst case scenario.

The maximum delay is the worst delay an advertiser had to experience, from the time it started transmission till the time it was discovered. This value increased as the number of advertisers increased. As per the simulation results, for the worst-case scenario if 100 advertisers started transmission at the same time then the maximum waiting time an advertiser had to experience was 65.107 s and this time was reduced to 40.50 s. The comparison of maximum delay with and without algorithm application can be seen in Figure 12.

The scenario where all the advertising devices enter their advertising state to transmit advertising PDUs to an initiator in a dense environment was simulated. This result depicts that if all advertising devices have an intention of connection setup then how long will they have to wait. This delay truly has an impact on the overall BLE connection setup latency. Figure 13 shows the results of the comparison of the total discovery time of all the advertisers in a dense environment when the discovery is done by the initiator. After the implementation of the algorithm, on average this value was decreased by 40.66%. Figure 14 shows the comparison of average delay latency. For 100 devices, in the ordinary scenario, the average value of an advertiser to be discovered by an initiator was 9.82 s, and after the implementation of the algorithm, this value was reduced to 4.477 s. In Figure 15, the comparison of maximum waiting time is done for before and after the algorithm implementation. For 100 devices the maximum time an advertiser had to wait before it received CONNECT_REQ was observed to be 94.56 s. This value was reduced to 59.12 s after the algorithm was applied.

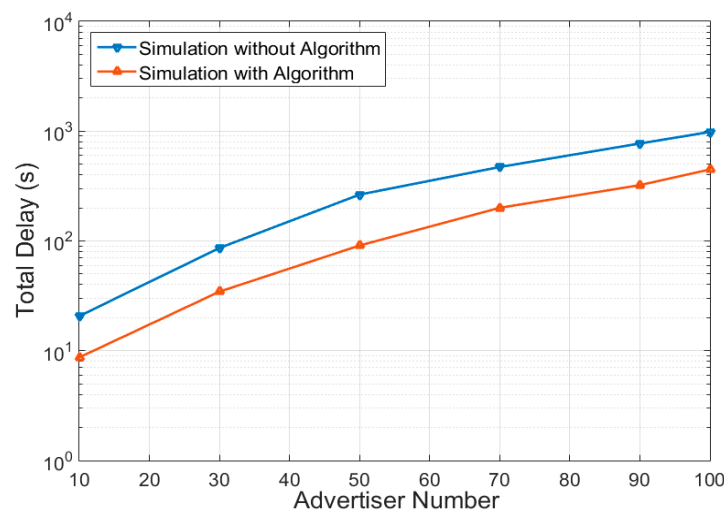


Figure 13. Total delay for advertiser discovery by the initiator in the worst-case scenario.

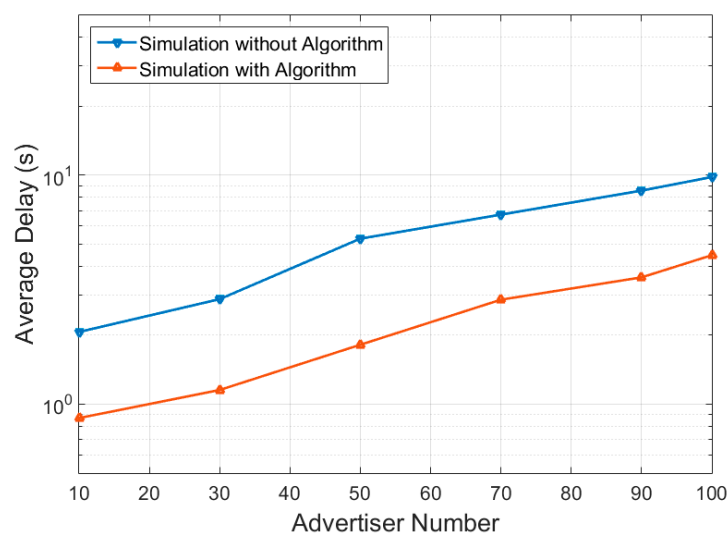


Figure 14. Average delay for advertiser discovery by the initiator in the worst-case scenario.

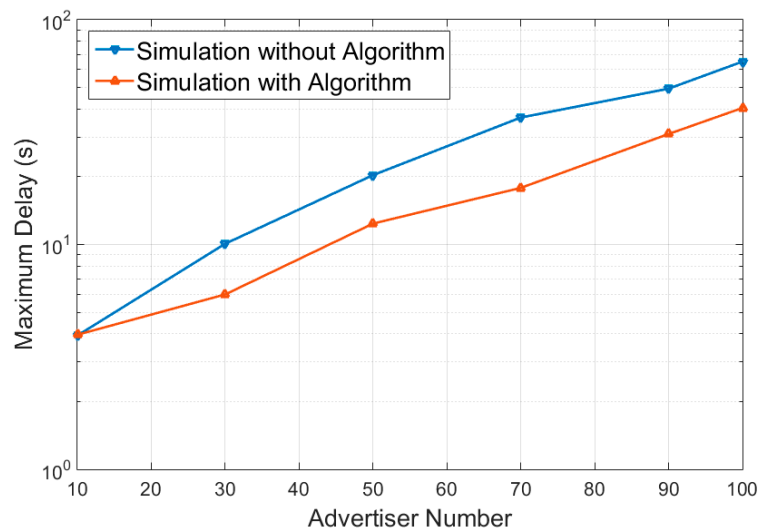


Figure 15. Maximum delay of one advertiser discovery by the initiator in the worst-case scenario.

The energy consumption calculation is summarized in Figure 16. The result shows the energy consumption profile by varying $T_{AdvInterval}$ for different values of N . As mentioned earlier for the energy consumption the model proposed in [25] was used. According to this model the energy consumption of an advertiser for one advertisement interval is given as 532.034 (mA × ms). In Figure 16 the energy consumption for the selected value of δ is observed as the minima. According to the results it can be inferred that when $T_{AdvInterval} = 200$ ms, the energy consumption is maximum. As the algorithm proceeds and arrives at the best value of δ the energy consumption decreases. The reason for high energy consumption for $T_{AdvInterval} = 200$ ms is the fact that when there is large number of N , more collisions will occur and hence the discovery process will take longer time and the energy consumption will increase. But as the best value of δ is achieved we observe the reduction in the energy consumption. After this we again can see there is a rise in the energy consumption until the maximum value of $T_{AdvInterval}$ is reached which is 10.24 s.

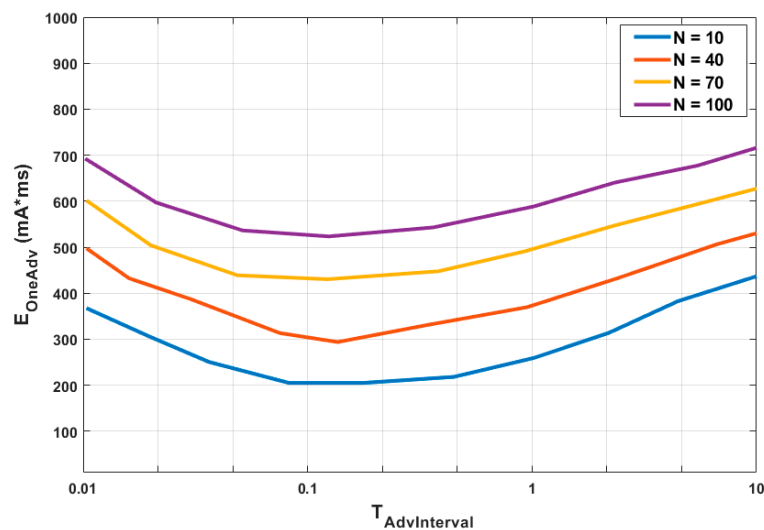


Figure 16. Energy consumption during advertisement event.

From all the results presented in this section, it can be concluded that the algorithm improved the delay of discovery in multiple setups. Depending upon the type of scanning device (i.e., scanner or initiator) and the density of devices in the vicinity, the improvement in average latency ranges from 35% to 55%. If the latency is improved, it implies that the system will end up being more energy

efficient. Because the devices would have to spend lesser time in scanning and looking for other devices. These results are also promising for applications where advertising devices and scanning devices are expected to exchange information without any prior knowledge of each other's existence.

5. Conclusions

BLE has become one of the most promising technology to be used in IoT applications. BLE is low energy, low cost, long lasting, and simple wireless technology that supplements its usage with IoT applications. As we are entering the 5th generation of wireless technology, the influx of smart devices and interconnected devices is exponentially increasing. This means for IoT services using BLE, as the number of devices increases in a surrounding the device discovery will take a long time as the probability of the collision will increase. In such a scenario, the scanner will have to discover all the surrounding devices in a very short time. The task of discovering the surrounding devices in a short time is very challenging. As the number of devices increases, the number of collisions also increases which in turn increases the discovery time. This paper proposes an algorithm which helps in reducing the discovery time of the advertisers in the surrounding. The proposed algorithm utilizes the freedom given in the BLE specifications for setting the parameters ranges. The proposed algorithm uses the information sent in the PDU by the advertisers to the scanner. Using that information, the scanner adjusts the scan interval and scan window time so that the device discovery time is reduced which in turn reduces the energy consumption. The results show us that by adjusting the parameters of scan interval and scan window time, the device discovery latency is reduced. Depending upon the type of scanning device (i.e., scanner or initiator) and the density of devices in the vicinity, the improvement in average latency ranges from 35% to 55%. The proposed algorithm further improves the results as it also uses the exchange of PDU from advertiser to a scanner which helps in managing the advertiser discovery time.

Author Contributions: The research presented in this paper was a collaborative effort among all authors. S.S.W. and F.I. conceived, implemented, and simulated the results along with the paper write-up. S.S.W., Y.S.L., and H.S.K. wrote the paper and discussed the results and revised the manuscript critically.

Funding: This work was funded by National Research Foundation of Korea (NRF), grant number 2016R1A2B4008457.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Jacob, E.; Astorga, J.; Unzilla, J.J.; Huarte, M.; García, D.; López-De-Lacalle, L.N. Towards a 5G compliant and flexible connected manufacturing facility|Hacia una infraestructura de fabricación flexible, conectada e integrable en redes 5G. *Dyna* **2018**, *93*, 656–662. [CrossRef]
- Barrio, H.G.; Morán, I.C.; Ealo, J.A.; Barrena, F.S.; Beldarrain, T.O.; Zabaljauregui, M.C.; Zabala, A.M.; Arriola, P.J.A.; de Lacalle Marcaide, L.N.L. Proceso De Mecanizado Fiable Mediante USO Intensivo Demodelizacion Y Monitorizacion Del Proceso: ENFOQUE 2025. *DYNA* **2018**, *93*, 689–696. [CrossRef]
- Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
- Want, R.; Schilit, B.N.; Jenson, S. Enabling the Internet of Things. *Computer* **2015**, *48*, 28–35. [CrossRef]
- Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. [CrossRef]
- Bluetooth Core Specification Version 4.2 (2014)*. Bluetooth SIG. Available online: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=441541 (accessed on 26 April 2019).
- Bluetooth Core Specification Version 5.1 (2019)*. Bluetooth SIG. Available online: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080 (accessed on 26 April 2019).
- Gomez, C.; Oller, J.; Paradells, J.; Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors* **2012**, *12*, 11734–11753. [CrossRef]

9. Collotta, M.; Pau, G. Bluetooth for Internet of Things: A fuzzy approach to improve power management in smart homes. *Comput. Electr. Eng.* **2015**, *44*, 137–152. [[CrossRef](#)]
10. Zhang, T.; Lu, J.; Hu, F.; Hao, Q. Bluetooth low energy for wearable sensor-based healthcare systems. In Proceedings of the 2014 IEEE Healthcare Innovation Conference (HIC), Seattle, WA, USA, 8–10 October 2014; pp. 251–254.
11. Martin, P.D.; Rushanan, M.; Tantillo, T.; Lehmann, C.U.; Rubin, A.D. Applications of Secure Location Sensing in Healthcare. In Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '16), Seattle, WA, USA, 2–5 October 2016; pp. 58–67.
12. Kang, H.-W.; Kim, C.-M.; Koh, S.-J. ISO/IEEE 11073-Based Healthcare Services over IoT Platform Using 6LoWPAN and BLE: Architecture and Experimentation. In Proceedings of the 2016 International Conference on Networking and Network Applications (NaNA), Hakodate City, Hokkaido, Japan, 23–25 July 2016; pp. 313–318.
13. Ramlee, R.A.; Tang, D.H.Z.; Ismail, M.M. Smart home system for Disabled People via Wireless Bluetooth. In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 11–12 September 2012; pp. 1–4.
14. Shan, G.; Park, B.; Nam, S.; Kim, B.; Roh, B.; Ko, Y.-B. A 3-dimensional triangulation scheme to improve the accuracy of indoor localization for IoT services. In Proceedings of the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), Victoria, BC, Canada, 24–26 August 2015; pp. 359–363.
15. Johnson, T.A.; Seeling, P. Localization using bluetooth device names. In Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '12), Hilton Head, SC, USA, 11–14 June 2012; p. 247.
16. Newman, N. Apple iBeacon technology briefing. *J. Direct Data Digit. Mark. Pract.* **2014**, *15*, 222–225. [[CrossRef](#)]
17. Decuir, J. Introducing Bluetooth Smart: Part II: Applications and updates. *IEEE Consum. Electron. Mag.* **2014**, *3*, 25–29. [[CrossRef](#)]
18. Liu, J.; Chen, C.; Ma, Y.; Xu, Y. Adaptive Device Discovery in Bluetooth Low Energy Networks. In Proceedings of the 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), Dresden, Germany, 2–5 June 2013; pp. 1–5.
19. Kim, J.; Han, K. Backoff scheme for crowded Bluetooth low energy networks. *IET Commun.* **2017**, *11*, 548–557. [[CrossRef](#)]
20. Shan, G.; Lee, B.; Shin, S.-H.; Roh, B. Design and implementation of simulator for analysis of BLE broadcast signal collision. In Proceedings of the 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 11–13 January 2017; pp. 448–452.
21. Balogh, A.; Imre, S. Simulation and analysis of concurrent BLE link layer state machines running within the same physical device. In Proceedings of the 2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2017), Split, Croatia, 21–23 September 2017; pp. 1–8.
22. Liu, J.; Chen, C.; Ma, Y. Modeling Neighbor Discovery in Bluetooth Low Energy Networks. *IEEE Commun. Lett.* **2012**, *16*, 1439–1441. [[CrossRef](#)]
23. Liu, J.; Chen, C.; Ma, Y. Modeling and performance analysis of device discovery in Bluetooth Low Energy networks. In Proceedings of the 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, USA, 3–7 December 2012; pp. 1538–1543.
24. Cho, K.; Park, W.; Hong, M.; Park, G.; Cho, W.; Seo, J.; Han, K. Analysis of latency performance of bluetooth low energy (BLE) networks. *Sensors* **2014**, *15*, 59–78. [[CrossRef](#)] [[PubMed](#)]
25. Jeon, W.S.; Dwijaksara, M.H.; Jeong, D.G. Performance Analysis of Neighbor Discovery Process in Bluetooth Low-Energy Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 1865–1871. [[CrossRef](#)]
26. Liu, J.; Chen, C.; Ma, Y.; Xu, Y. Energy Analysis of Device Discovery for Bluetooth Low Energy. In Proceedings of the 2013 IEEE 78th Vehicular Technology Conference (VTC Fall), Las Vegas, NV, USA, 2–5 September 2013; pp. 1–5.
27. Schrader, R.; Ax, T.; Rohrig, C.; Fuhner, C. Advertising power consumption of bluetooth low energy systems. In Proceedings of the 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Offenburg, Germany, 26–27 September 2016; pp. 62–68.

28. Silva, S.; Soares, S.; Fernandes, T.; Valente, A.; Moreira, A. Coexistence and interference tests on a Bluetooth Low Energy front-end. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 1014–1018.
29. Siekkinen, M.; Hienkari, M.; Nurminen, J.K.; Nieminen, J. How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Paris, France, 1 April 2012; pp. 232–237.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).