

## BLE Parameter Optimization for IoT Applications

Andreina Liendo, Dominique Morche, Roberto Guizzetti, Franck Rousseau

► **To cite this version:**

Andreina Liendo, Dominique Morche, Roberto Guizzetti, Franck Rousseau. BLE Parameter Optimization for IoT Applications. IEEE International Conference on Communications (ICC'2018), May 2018, Kansas City, MO, United States. <hal-01775056>

**HAL Id: hal-01775056**

**<https://hal.archives-ouvertes.fr/hal-01775056>**

Submitted on 24 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# BLE Parameter Optimization for IoT Applications

Andreina Liendo<sup>\*‡¶</sup>, Dominique Morche<sup>‡</sup>, Roberto Guizzetti<sup>¶</sup> and Franck Rousseau<sup>\*</sup>

<sup>\*</sup>Grenoble Alps University, Grenoble Institute of Technology, France

Email: firstname.lastname@imag.fr

<sup>‡</sup>CEA-LETI, MINATEC, 38000 Grenoble, France

Email: firstname.lastname@cea.fr

<sup>¶</sup>STMicroelectronics, 38920 Crolles, France

Email: firstname.lastname@st.com

**Abstract**—Bluetooth Low Energy (BLE) has been designed as a power efficient protocol for small portable and autonomous devices, showing its efficiency for connecting these devices with smartphones to periodically and frequently exchange data, like heart rate or notifications. Additionally, BLE is present in almost every smartphone, turning it into perfect ubiquitous remote control for smart homes, buildings or cities. Nevertheless, there is still room to improve BLE performance for typical IoT use cases where battery lifetime should reach several years. In this paper we propose an extension to a model for evaluating BLE performance, latency and energy consumption, in order to provide realistic results based on various scenario conditions. In addition, we propose a parameter optimization of the BLE Neighbor Discovery process, in order to obtain the best performance possible depending on the constraints of specific use cases. Our results on two typical IoT test-cases show that advertiser battery lifetime can be increased up to  $\approx 89\times$  (9.55 days to 2.32 years) for a first case, and  $\approx 281\times$  (9.55 months to 7.36 years) for a second case.

**Index Terms**—Internet of Things; Bluetooth Low Energy; Neighbor Discovery; Discovery Latency; Energy Consumption; Battery Lifetime; Testbed; Measurements.

## I. INTRODUCTION

BLE has two communication modes according to its specification: Neighbor Discovery or ND and Connected Mode or CM. ND is the first phase, where two devices discover each other to establish a link. CM is the second phase, where devices can exchange data over their connection, which is teared down once the session is over. While in ND, the central device is the *scanner* and the peripheral device is the *advertiser*. When reaching CM, the central device becomes the *master* and peripheral the *slave*. During ND, communication is asynchronous and devices transit through 4 of the 5 BLE states [1]: *standby*, *scanning* and *initiating* for scanners, or *standby* and *advertising* for advertisers, using only the 3 advertising channels. When in CM, a device transits through *connection* and *standby* using the 37 data channels. The time spent in CM depends on the size of the data and/or the duration of the session, whereas the time a BLE transceiver will spend transmitting, receiving or in standby mode during ND is determined by the parameters given in Table I. The values assigned to these parameters determine two important and interdependent characteristics of the ND phase for BLE devices that have a huge impact on user experience and device lifetime: Discovery Latency (DL)

and energy consumption. DL is the time it takes for a device to discover or to be discovered, in other words the time to establish a communication.

Table I: BLE Neighbor Discovery parameters

Name	Notation	Value according to the standard
Scan Interval	$T_{SI}$	$\leq 10.24$ s
Scan Window	$T_{SW}$	$\leq T_{SI}$
Advertising Interval	$T_{AI}$	$20$ (or $100$ ) ms $\leq T_{AI} \leq 10.24$ s integer multiple of $0.625$ ms
Advertising Delay	$\rho$	pseudo-random value in $[0, 10]$ ms
Advertising Event	$T_{AE}$	$= T_{AI} + \rho$
Adv. Indication interval	$T_{AW}$	$\leq 10$ ms (low duty cycle)

BLE is suitable for a very wide range of application scenarios, but to maximize the lifetime of the devices while guaranteeing the user experience expected for a given application, it is necessary to find a suitable set of parameters that are specific to the type of communication patterns being used and the constraints specific to the application. In this work, we propose a method to find optimal sets of parameters to minimize the energy consumption for a wide range of BLE applications, thus maximizing battery lifetime.

In previous work [2]–[6] authors have proposed models for energy consumption during ND from the advertiser’s perspective and considering that the scanner is already listening when the advertiser starts advertising. These models are based on quantitative models of the physical layer via BLE SoC measurements of current consumption [2], and probabilistic models of the Link Layer of the protocol [3]–[6]. For certain use cases the role of the devices might be reversed, for example in the case where there are fixed advertisers and scanners are passing by. In this situation, the advertiser is waiting for a scanner, which is not taken into account in existing models.

In this paper, we present an extension and optimization of the model proposed by Kindt *et al.* [4] in Section III. Then we propose a method for parameter optimization in Section IV. The objective is to obtain the best BLE performance possible during ND, with respect to two important metrics: Discovery Latency (DL) and energy consumption. Based on our parameter optimization, we give DL, energy consumption and battery lifetime results for two IoT use cases in Section VI. These results are compared with performance obtained when using configurations recommended by the Bluetooth SIG Profiles

[7]. DL model and parameter optimization, are validated via testbed experiments in Section V.

## II. OPTIMIZING BLE FOR A WIDE RANGE OF SCENARIOS

The BLE standard supports a wide range of parameter values for ND. The choice of these parameters directly affect the performances of BLE devices. Therefore, it is of capital importance to use an optimal parameter configuration in order to ensure a fair tradeoff between DL and energy consumption. According to the BLE specification, before establishing a connection, the scanner must first listen to the advertising channel during the ND in order to synchronize with the advertiser. The scanner/master is the initiator and is responsible for establishing a connection with one or several advertisers/slaves. Once in Connected Mode, or CM, master and slave will periodically exchange application data. From the point of view of BLE communication, the scanner/master is the central device whereas the advertiser is the peripheral which is typically at the sensor side. However, for certain use cases, the scanner may play the peripheral role at the application level.

Applications based on proximity sensing are an example where the advertiser is configured to broadcast data to advertise periodically. Multiple scanners in range can receive data without the need for a connection establishment. From the application level point of view, the advertiser and the scanner would play the central and the peripheral role respectively, and furthermore, the scanner is at the user side, meaning that for such use cases, DL and energy consumption criticality are at the scanner side. This shows that efforts to improve performance should be focused on one side or another, depending on constraints that are imposed by the use case.

The Bluetooth SIG has given a list of profiles [7] intended to provide efficient configurations for BLE devices for different applications, such as collecting sensor information, health, sports and fitness, environmental sensing and proximity. These profiles define the behavior for both central and peripheral devices, and they include recommended scanner/advertiser configurations to ensure optimal DL and energy consumption during ND. There are 24 profiles in total, where the same configuration is recommended for 10 different profiles and it is depicted in Table II, 4 profiles recommend different configurations, 6 profiles state that devices should be configured with consideration for user expectations of connection establishment, and the rest state that connection establishment requirements are defined by a higher layer specification.

To the best of our knowledge, there is no performance evaluation using these profiles and there is no evaluation considering the scanner as a peripheral device with constrained DL and energy consumption. The analytical model proposed by Kindt *et al.* [4] can be used to determine the right configuration depending on application constraints. But results from this kind of model are very general and might require very long computation time. Getting results from Algorithm 1 can take from minutes to several hours when evaluating configurations that uses large parameters. Furthermore, this model is not

suitable when the scanner behaves as the peripheral device at the application level.

We extend and optimize this model to estimate BLE performance, DL and energy consumption, regardless of the role of the devices. In addition, we propose a parameter optimization method based on this model, in order to obtain the lowest BLE energy consumption possible, while taking into account use case requirements such as maximum acceptable Discovery Latency. Finally, we evaluate the Bluetooth SIG Profiles for some typical IoT use cases and compare with the results we find with our new model. We present the results in terms of battery lifetime based on the battery model presented in our other work [8].

Table II: Recommended Advertising Interval, Scan Interval and Scan Window Values (Table 5.1 and 5.2 in [9])

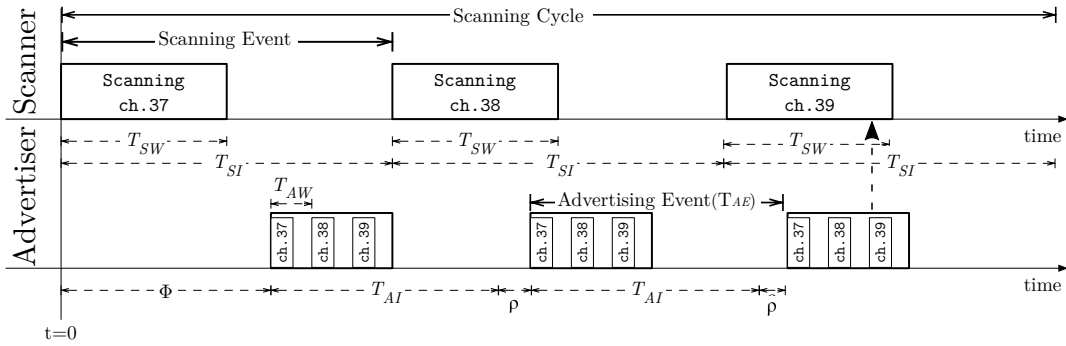
Advertising/Scan Duration	Parameter	Value
First 30 seconds (fast connection)	Adv. Interval	20 ms to 30 ms
After 30 seconds (reduced power)	Adv. Interval	1 s to 2.5 s
First 30 seconds (fast connection)	Scan Interval	30 ms to 60 ms
	Scan Window	30 ms
After 30 seconds (reduced power)	Scan Interval	1.28 s
– Option 1	Scan Window	11.25 ms
After 30 seconds (reduced power)	Scan Interval	2.56 s
– Option 2	Scan Window	11.25 ms

## III. MODEL OPTIMIZATION

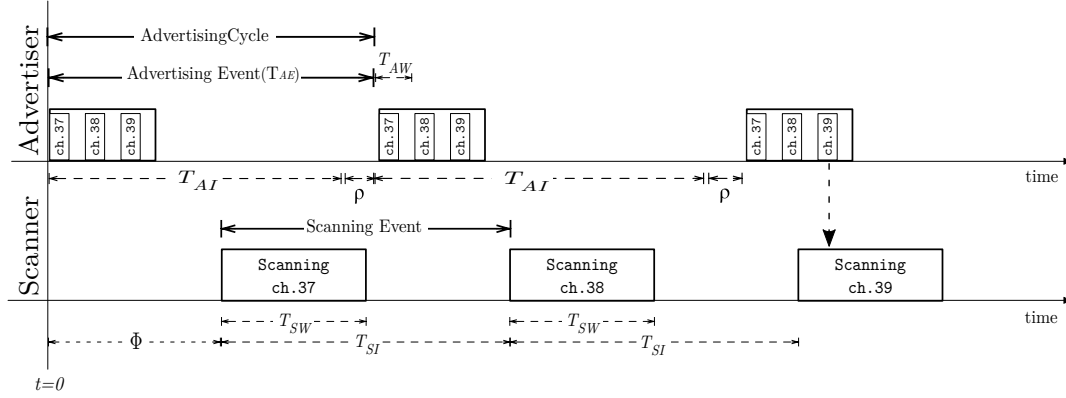
We based our work on the more complete and precise model available in the literature [4]. The model considers that the advertiser starts at a given phase offset called  $\phi$  after  $t = 0$ , moment at which the scanner starts scanning on channel 37 as shown in Figure 1a. In order to determine scanner performance when constrained DL and energy consumption is at the scanner side, we propose to consider the case where the advertiser starts advertising at a given phase offset  $\phi$  before  $t = 0$ . In other words, in this case, the scanner starts scanning at a random time within one advertising cycle as illustrated in Figure 1b.

Algorithm 1 from [4], is used to estimate the advertiser DL. It is the elapsed time between the start of the first advertising event, until the start of the advertising event received by the scanner. The duration of an advertising event successfully received by the scanner  $t_{advEvent}(ch)$  is added to obtain the total advertiser DL. It depends on the channel over which reception takes place, which is estimated at line 9.

The condition to consider a successful reception is that the starting point of an advertiser event  $n$  falls into a region within a scanner interval called *effective scanning window* which goes from  $t_{semin}(k)$  to  $t_{semax}(k)$ . The effective scanning window is different for each of the advertising channels and depends on four parameters: scanning interval  $T_{SI}$ , scanning window  $T_{SW}$ , duration of an advertising packet transmission  $T_{apk}$  and the duration of hopping to the next channel  $t_{ch}$  (see Figure 1, where  $T_{AW} = T_{apk} + t_{ch}$ ). The effective scanning window is estimated as seen in Table III, where *tearly* is the time with respect to each advertising channel from which, if



(a) The Scanner is available waiting for an advertising packet



(b) The Advertiser is available waiting to be discovered by the scanner

Figure 1

an advertising event  $n$  begins, the packet can be successfully received even if  $tearly$  is not within the scanning window.  $tlate$  is the time with respect to each advertising channel from which, if an advertising event  $n$  begins, the packet can't be successfully received, even if  $tlate$  is within the scanning window. For more details we refer the reader to the paper [4].

The probability of an advertising event having started between  $t_{semin}(k)$  and  $t_{semax}(k)$  is called  $P_{hit}$ , it is calculated at line 10 of Algorithm 1, where  $n$  and  $k$  are the current evaluated advertiser and scanner events which are considered according to lines 13 and 6 respectively.  $PcM$  is the probability that the  $n_{th}$  advertising events don't lead to a successful reception (cumulative miss probability). With increasing values of  $n$ , the probability that one of the advertising events is received successfully, grows. Thus,  $PcM$  shrinks with growing  $n$ . The algorithm terminates if  $(1 - PcM)$  is smaller than a lower bound  $\epsilon = 0.9999$ . The expected DL for a given  $\phi$  offset is calculated at line 11. It is considered for the expected value of  $\rho$  to be  $5ms$  and the error is neglected. Then the average DL for a given  $T_{SI}, T_{SW}, T_{AI}$  configuration is obtained [4] by integrating the expected DL results over all possible values of  $\phi$ . A numerical integration is then performed by multiplying the results with  $\Delta$  and computing the sum of these values.

For calculating the probability  $P_{hit}$  of whether an advertising event is successfully received, the probability density

function (PDF) of the start of an advertising event over time  $t$  (which depends on both  $n$  and  $\phi$ ) is required. The shape of the distribution depends on  $n$ . For more detail on the shape of the distribution we refer the reader to the paper [4].

Then the consumed charge in Coulombs during DL can be calculated knowing the current and voltage of the circuit for each event. The current and voltage vary from one manufacturer to the other and are extracted from quantitative models of the physical layer which are based on SoC measurements [2]. Finally, knowing the charge and DL, energy consumed during DL can be estimated. Quantitative models of physical layer in our work are based on TI CC2540 and ST BlueNRG current consumption measurements extracted from technical literature [10], [11].

Table III: Effective Scanning Window

Ch	$tearly$	$tlate$
37	0	$T_{apk}$
38	$T_{apk} + t_{ch}$	$2T_{apk} + t_{ch}$
39	$2T_{apk} + 2t_{ch}$	$3T_{apk} + 2t_{ch}$
Ch	$t_{semin}(k)$	$t_{semax}(k)$
37	0	$(k-1)T_{SI} + T_{SW} - tlate$
38	$(k-1)T_{SI} - tearly$	$(k-1)T_{SI} + T_{SW} - tlate$
39	$(k-1)T_{SI} - tearly$	$(k-1)T_{SI} + T_{SW} - tlate$

In order to evaluate DL from scanner perspective when it is

---

**Algorithm 1** BLE Average Advertiser DL

---

```
1:  $DL_{adv} \leftarrow 0$ 
2: for  $\phi = 0$  to  $3T_{SI}$  step  $\Delta$  do
3:    $n \leftarrow 0, DL_{exp} \leftarrow 0, P_{hit} \leftarrow 0, PcM \leftarrow 1, ch \leftarrow 37$ 
4:   while  $1 - PcM \leq \varepsilon$  do
5:      $t_{ideal} \leftarrow \phi + nT_{AI}, P_{hit} \leftarrow 0$ 
6:      $K_{min} = \lfloor \frac{t_{ideal}}{T_{SI}} \rfloor, K_{max} = \lfloor \frac{t_{ideal} + n5ms}{T_{SI}} \rfloor$ 
7:     for  $K = K_{min}$  to  $K = K_{max}$  do
8:        $ch \leftarrow \text{mod}(j, 3)$ 
9:        $(t_{semin}(k), t_{semax}(k)) \leftarrow \text{getInterval}(ch)$ 
10:       $P_{hit} \leftarrow P_{hit} + pk(t_{real}, n, \rho \sqrt{\frac{n}{12}}, t_{ideal}, t_{semin}(k), t_{semax}(k), \rho)$ 
11:       $DL_{exp} \leftarrow DL_{exp} + pk.PcM(n(T_{AI} + 5ms) + t_{advEvent}(ch))$ 
12:    end for
13:     $PcM \leftarrow PcM(1 - P_{hit}), n \leftarrow n + 1$ 
14:  end while
15:   $DL_{adv} \leftarrow DL_{adv} + DL_{exp}$ 
16: end for
17:  $DL_{adv} \leftarrow \frac{DL_{adv}}{\Delta}$ 
```

---

peripheral at the application level, we propose an optimization as depicted in Algorithm 2. Unlike in Algorithm 1, the phase offset  $\phi$  is within a negative range, that is the case where the advertiser starts advertising before the scanner starts scanning. We consider the advertiser starts a maximum of  $T_{AI} + \rho$  time before  $t = 0$  which is a full advertising cycle plus the maximum value of  $\rho$  as shown in Figure 1b. Before  $t = 0$  any advertising event  $n$  has 0 probability of being received by the scanner, so  $PcM$  is increased to its maximum value and the next event  $n$  is evaluated as shown in line 5 of Algorithm 2. The average DL has been computed such that all data related a specific event can be accessed once the algorithm has been executed as seen in line 18 of Algorithm 2. The foregoing aims to evaluate not only average but also worst case for parameter optimization purposes, as it will be discussed in Section IV, so we don't make use of  $DL_{adv}$  and  $DL_{exp}$  variables shown in Algorithm 1. Finally, average DL is calculated at line 19.

Algorithm 2 can be used for both cases: when analyzing BLE performance from the advertiser or scanner perspective. The main difference between these two cases lies in the definition of  $\phi$ . When analyzing performance from the advertiser perspective we use  $\phi = [0, 3T_{SI}]$  and when analyzing performance from the scanner perspective  $\phi = [-(T_{AI} + 10ms), 0]$  accordingly with Figure 1. For the sake of simplicity, henceforth when we refer to the performance from the advertiser point of view, we will refer to Algorithm 1, and we'll refer to Algorithm 2 when analyzing from scanner perspective. As it has been stated in [4], the model must be calibrated. Its precision lies in the choice of  $\Delta$  ( $\phi$  step). We have adjusted  $\Delta$  for Algorithm 1 according to Eq. 1 and for Algorithm 2 according to Eq. 2. The accuracy of the estimated advertiser and scanner DL, for both average and worst case values, are experimentally validated as explained in Section V.

$$\Delta = \begin{cases} 1ms & \text{if } 30ms \geq T_{SI} \leq 100ms \\ 5ms & \text{if } 100ms < T_{SI} < 640ms \\ 93.6ms & \text{if } T_{SI} \geq 640ms \end{cases} \quad (1)$$

$$\Delta = \begin{cases} 10\mu s & \text{if } 20ms \geq T_{AI} \leq T_{SW} + \rho \\ 19.9\mu s & \text{if } T_{SW} + \rho < T_{SI} < 6s \\ 29.9\mu s & \text{if } T_{SI} \geq 6s \end{cases} \quad (2)$$

---

**Algorithm 2** Proposed BLE Average Scanner DL

---

```
1: for  $\phi = -T_{AI} + 10ms$  to 0 step  $\Delta$  do
2:    $n = 1, PcM = 1, \varepsilon = 0.9999$ 
3:   while  $1 - PcM \leq \varepsilon$  do
4:      $t_{ideal} = \phi + (n - 1)T_{AI}, t_{real} = t_{ideal} + (n - 1)\frac{\rho}{2}$ 
5:     if  $t_{ideal} < 0$  then
6:        $PcM = 1, n = n + 1$ 
7:       continue
8:     end if
9:      $K_{min} = \lfloor \frac{t_{ideal}}{T_{SI}} \rfloor, K_{max} = \lceil \frac{t_{ideal} + n5ms}{T_{SI}} \rceil, P_{hit} = 0$ 
10:    for  $K = K_{min}$  to  $K = K_{max}$  do
11:       $t_{semin}(k, T_{SI}, T_{SW}, Tapk, tch)$ 
12:       $t_{semax}(k, T_{SI}, T_{SW}, Tapk, tch)$ 
13:       $pk(t_{real}, n, \rho \sqrt{\frac{n}{12}}, t_{ideal}, t_{semin}(k), t_{semax}(k), \rho)$ 
14:       $P_{hit} = P_{hit} + pk$ 
15:    end for
16:     $PcM = PcM(1 - P_{hit}), n = n + 1$ 
17:  end while
18:  $adv_{t_{real}} = t_{real}(\max(p_k)) + \text{abs}(\phi), scan_{t_{real}} = t_{real}(\max(p_k))$ 
19:  $advDL = \text{mean}(adv_{t_{real}}), scanDL = \text{mean}(scan_{t_{real}})$ 
```

---

#### IV. BLE PARAMETER OPTIMIZATION

BLE parameter optimization is achieved using a look-up table based simulation in Matlab using Algorithm 2. For evaluation of DL, the typical procedure is to fix two parameters, usually  $T_{SI}$  and  $T_{SW}$ , and to vary the other one,  $T_{AI}$ . As mentioned previously, this procedure can take a lot of time. We have executed the algorithm for typical BLE parameter values and all their possible combinations and stored them in a look-up table in order to save computing time when evaluating use case.

We sweep into the look-up table to determine the  $(T_{SI}, T_{SW}, T_{AI})$  configuration whose worst DL is no longer than a *critical latency* (CL) given by the use case: the maximum communication latency that guarantees acceptable operation or user experience. We have set up the simulation such that, the worst case DL has a maximum difference of 5 ms below the desired value (CL). Several configurations may satisfy a given latency requirement, for example, for a required advertiser DL of 200 ms at least a dozen of configurations can be found, with a scanner duty cycle ranging from 65% to 100%, using  $T_{SI} \leq 400$  ms and an advertiser duty cycle of up to  $\approx 5\%$ . In this case, we select the configuration that provides the smallest energy consumption on the advertiser side. Similarly, several configuration may satisfy a given latency requirement when constrained DL is at the scanner side, and we selected the configuration that provides the smallest energy consumption for the scanner.

In Table IV we give a set of parameters  $(T_{SI}, T_{SW}, T_{AI})$  obtained for a required latency CL between 200 ms and 1 s, when evaluating BLE from the advertiser perspective (advCL) and from scanner perspective (scaCL).

#### V. EXPERIMENTAL MODEL VALIDATION

##### A. Testbed Setup

We have validated the proposed model on a real testbed developed on top of WalT. WalT is a cheap, reproducible and highly configurable platform for network experiments developed at Univ. Grenoble Alpes [12]. It is basically composed

Table IV: Parameter Optimization Respect to CL

CL	Respect to advCL			Respect to scanCL	
	$T_{AI}$	$T_{SI}$	$T_{SW}$	$T_{SI}$	$T_{SW}$
200 ms	190 ms	400 ms	300 ms	400 ms	300 ms
300 ms	290 ms	500 ms	400 ms	600 ms	400 ms
400 ms	390 ms	60 ms	50 ms	800 ms	500 ms
500 ms	490 ms	500 ms	500 ms	900 ms	600 ms
600 ms	590 ms	300 ms	300 ms	1.28 s	640 ms
700 ms	690 ms	200 ms	200 ms	2.56 s	1.28 s
800 ms	790 ms	70 ms	60 ms	3.2 s	1.28 s
900 ms	890 ms	2.56 s	1.92 ms	3.84 s	1.28 s
1 s	990 ms	1.92 ms	1.28 ms	4.48 s	1.28 s

of 6 different elements: clients from which to access the server and the nodes, the server which is the brain of the platform, which interacts with nodes on which we run or drive experiments, interconnected with PoE switches to manage the network and restart nodes, and finally the docker hub on which we store WaT images containing full reproducible experiments. In our case, nodes are Raspberry Pis with one Asus USB-BT400 Bluetooth device attached, compatible with Bluetooth 4.0 Core, running Debian with the Linux BlueZ stack.

We have configured two nodes to play advertiser and scanner role. They both have a set of scripts that drive the communication between the two BLE devices. One task runs on the advertiser and starts the scanner through the wired network with a delay, before or after it itself starts advertising, depending on the scenario. The advertiser contains all the data related to  $T_{SI} - T_{SW} - T_{AI}$  configurations and iterates over  $T_{AI}$  with a given step, repeating every measurement a configurable number of times (100 in this case). While running the experiments we capture all the timestamped Bluetooth traffic with `tshark` for further analysis, mainly extracting the discovery latency. Docker images containing all the software to run the exact same experiment will be made available.

### B. Experimental Results

Figure 2 and 3 show the results obtained on the testbed compared to results obtained by simulation, using respectively Algorithm 1 with the modifications mentioned in Section III and Algorithm 2, for  $T_{SI} = 400$  ms and  $T_{SW} = 300$  ms. For the sake of legibility, we show detailed statistics only for testbed results and only the mean for simulation. In Figure 2 the advertiser is peripheral, and in Figure 3 the scanner is peripheral. High average DL is obtained near the values of  $T_{AI}$  that are multiples of  $T_{SI}$ , corresponding to the peaks observed in both figures, as it was observed in previous work too [4]. Around these values, if an advertising event starts between the end of the scanning window and the beginning of the next scanning event, then the next advertising event is very likely to be missed, as the random delay  $\rho$  is not enough to compensate the difference between  $T_{SI}$  and  $T_{SW}$ . The peaks should be avoided in configurations, as very large average and worst case DL are obtained for these configurations, resulting in higher energy consumption and bad user experience. Experimental

results show the same pattern as the results obtained from simulations of the model. As said earlier, details statistics for simulation results were omitted, since they fit perfectly to experimental results, except for extreme mean and max values at the peaks, they are just more regular as you would expect since they were obtained from an ideal system, with a very large number of trials. In both cases, the peaks are exactly at the same location, which means that the model and experimentation behave similarly. However, we can see that there is a significant difference in the extreme values. This can be explained easily by the fact that we drive experiments from user space, where high precision timing is impossible to achieve. This introduces a high variability in the results where the curve is very steep since a small imprecision in the timing, can result in a large variation in the measured delay. We notice on the graph that the distribution of values is very asymmetric, the median stays low, but the mean grows very quickly at the peaks, the 1<sup>st</sup> and 3<sup>rd</sup> quartiles are close, but the 0.05 and 0.95 quantiles show much more variability: there are few occurrences where the DL is very large but they have a very big impact. Also, these occurrences are catastrophic for the user experience, we reached a worst case scanner discovery latency of 255 s in our experiments, more than 4 min!

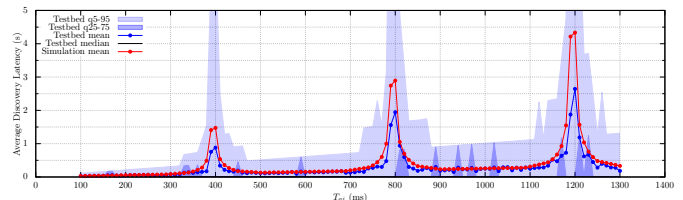


Figure 2: Advertiser Discovery Latency with Algorithm 1

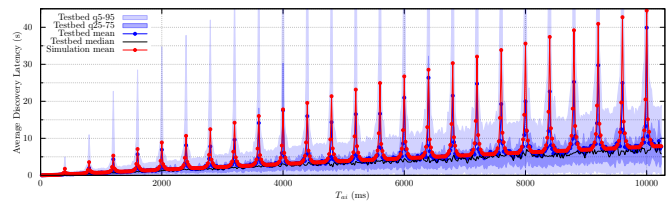


Figure 3: Scanner Discovery Latency with Algorithm 2

Once the model has been validated we proceed to validate our parameter optimization method. We have tested the configurations contained in Table IV which, as mentioned in Section IV, are based on the worst DL of a  $T_{AI}-T_{SI}-T_{SW}$  configuration, such that it does not exceed a specific CL. Table V and VI show the resulting worst DL obtained from both simulation and experiments, corresponding to the configurations that provide best energy consumption on the advertiser side (Table V) and scanner side (Table VI). As it can be seen, results are very accurate with a maximum error of  $\approx 3\%$  of our simulations compared to the experimental results. This validates our DL model and parameter optimization, and thus energy consumption for any BLE manufacturer.

Table V: Parameter Optimization respect to advertiser CL

CL	$T_{SI}$	$T_{SW}$	$T_{AI}$	simul.	expe.	error
200 ms	400 ms	300 ms	190 ms	198.1 ms	204.40 ms	3.08 %
300 ms	500 ms	400 ms	290 ms	298.1 ms	303.51 ms	1.78 %
400 ms	60 ms	50 ms	390 ms	398.1 ms	406.02 ms	1.95 %
500 ms	500 ms	500 ms	490 ms	498.1 ms	505.82 ms	1.53 %
600 ms	300 ms	300 ms	590 ms	598.1 ms	596.31 ms	0.30 %
700 ms	200 ms	200 ms	690 ms	698.1 ms	706.19 ms	1.15 %
800 ms	70 ms	60 ms	790 ms	798.1 ms	803.81 ms	0.70 %
900 ms	2.56 s	1.92 ms	890 ms	898.1 ms	905.61 ms	0.83 %
1 s	1.92 ms	1.28 ms	990 ms	998.1 ms	1.006 s	0.79 %

Table VI: Parameter Optimization respect to scanner CL

CL	$T_{SI}$	$T_{SW}$	$T_{AI}$	simul.	expe.	error
200 ms	400 ms	300 ms	190 ms	204.2 ms	200.76 ms	1.68 %
300 ms	600 ms	400 ms	290 ms	304.2 ms	306.76 ms	0.83 %
400 ms	800 ms	500 ms	390 ms	404.2 ms	403.79 ms	0.10 %
500 ms	900 ms	600 ms	490 ms	504.2 ms	504.82 ms	0.12 %
600 ms	1.28 s	640 ms	590 ms	604.2 ms	601.83 ms	0.39 %
700 ms	2.56 s	1.28 s	690 ms	704.2 ms	688.89 ms	2.17 %
800 ms	3.2 s	1.28 s	790 ms	804.2 ms	801.84 ms	0.29 %
900 ms	3.84 s	1.28 s	890 ms	904.2 ms	892.81 ms	1.26 %
1 s	4.48 s	1.28 s	990 ms	1004.2 s	987.87 ms	1.63 %

## VI. APPLICATION TO REALISTIC SCENARIOS

In order to provide results for typical IoT scenarios based on our parameter optimization, we propose two test cases and compare them with results when using recommended SIG Profiles configurations, more specifically the Proximity Profile [9]. These two test cases are a retail store and a medical telemetry system which are described below.

### A. Retail store use case

This test case targets iBeacon technology (a Proximity application). A device which generates iBeacon advertisements is called beacon. Beacons establish a region around them by iBeacon signals as shown in Figure 4. A device supporting an iBeacon application can determine if it has entered or exited from the region, and can approximate its distance to the beacon via signal strengths. Beacons transmit advertisement data frames containing information about what’s on sale so the user can find the item and receive extra information about it. Supposing the user is walking around the beacon, the time to pass near it must be considered to ensure a connection between the beacon (advertiser) and the smartphone (scanner); so the user can successfully receive the notification.

In the worst case, where the user walks by 29.5m far from the beacon, for a total walking distance of 10.9m as shown in Figure 4, the user has a maximum time to establish a connection of 11.22 s if the user walks at speed of 3.5 km/h [13] or 4.36 s at a speed of 9 km/h [14]. The user passes through the region at a constant speed in a straight line without stopping. BLE on the user’s smartphone is active at the moment to enter the region. For performance evaluation, we suppose ideal channel conditions and consider the worst case.

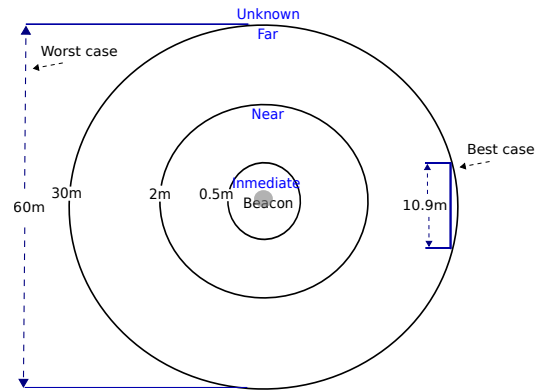


Figure 4: iBeacon regions. Worst case and best case walking distance around iBeacon

In accordance with the above mentioned, we have considered a maximum acceptable latency CL of 4.36s. In addition, with the aim of providing reliability in case of packet loss where an entire period would have to pass before the packet can be received, we consider  $CL = 2.18s$ , thus ensuring that the packet can be received while the user is still in range with the beacon. This is the CL reference value used for parameter optimization. The chosen configuration is shown in Table VII.

Since the advertiser is considered to be active permanently, we do not consider dynamic reconfiguration after 30 s as in the SIG Profiles. We evaluate performance of this use case based on Algorithm 2, where the advertiser is considered to be available waiting to be discovered by the scanner. Table VII shows the simulation results for average energy consumption ( $E_{avg}$ ), average DL ( $DL_{avg}$ ), worst case DL ( $DL_{wc}$ ) and lifetime for advertiser and scanner of a TI CC2540 and a ST BlueNRG. These values represent the typical expense during one connection between advertiser and scanner during ND. Our proposed configuration meets the requirements of the use case, while implementing a duty cycle that provides a longer lifetime for both scanner and advertiser.

Using the CC2540 device from TI, advertiser (beacon) lifetime can be 105 times higher, compared with recommended configurations from the Bluetooth SIG. At the same time providing a scanner DL that satisfies application requirements and providing a good user experience. When using ST BlueNRG device, advertiser lifetime can be 89 times higher using our proposed configuration compared to the Bluetooth SIG and scanner lifetime can be twice higher with our proposition for both TI and ST devices. Scanner results are obtained based on a 225mAh coin cell battery, and although the scanner is expected to be running on a smartphone which has higher capacity, this proves that smartphone battery usage can be more efficient. Additionally, we can see that using the Bluetooth SIG profile, it would be necessary to connect the beacon to a permanent source of energy. In contrast, with our proposition, the beacon can operate for more than 2 years on a single coin cell battery when using ST BlueNRG, which achieves the longest lifetime.

Table VII: Retail store use case DL, energy consumption and lifetime results

SIG configurations	TI		BlueNRG		
	Advertiser	Scanner	Advertiser	Scanner	
$T_{SI} = 60ms, T_{SW} = 30ms$ $T_{AI} = 20ms$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	231.16 $\mu$ J 34.9ms 51.6ms 3.82 days	1.1mJ 22.5ms 34.2ms 1.76 days	66.711 $\mu$ J 34ms 50.7ms 9.55 days	452.11 $\mu$ J 20ms 31.7ms 2.56 days
$T_{SI} = 60ms, T_{SW} = 30ms$ $T_{AI} = 30ms$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	202.38 $\mu$ J 41ms 71.6ms 5.54 days	1.24mJ 23.6ms 39.2ms 1.76 days	57.78 $\mu$ J 40.1ms 70.7ms 13.82 days	476.78 $\mu$ J 21.1ms 36.6ms 2.56 days
<b>Proposed in this work</b>	<b>Advertiser</b>	<b>Scanner</b>	<b>Advertiser</b>	<b>Scanner</b>	
$T_{SI} = 10.24s, T_{SW} = 2.56s$ $T_{AI} = 2.2s$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	192.11 $\mu$ J 2.17s 4.41s 1.1 years	58.4mJ 1.11s 2.21s 4.05 days	60.07 $\mu$ J 2.22s 4.41s 2.32 years	25.7mJ 1.11s 2.21s 5.62 days

### B. Medical telemetry system use case

We now analyze a medical telemetry use case where an in-body implanted pacemaker uses BLE to connect to a mobile device or other access point supporting BLE, to transmit information to a medical institution. The implanted sensor operates on a 225 mAh coin cell battery and plays the beacon or peripheral role (advertiser), so the challenge is to extend battery lifetime as long as possible, as it has to remain active permanently. A connection is required to collect data from the pacemaker to the smartphone or other mobile device via BLE. The key parameters to be evaluated are the connection time on the user side and its energy consumption during ND. The scanner is active when sending data to the hospital is required, whose frequency can range from every 3 months to every 12 months [15]. The scanner is in range at the moment of connection. We base the parameter optimization and performance evaluation on Algorithm 1, since the advertiser plays the peripheral role at the application level and has constrained DL and energy consumption. We suppose ideal channel conditions.

Table VIII shows DL, energy consumption and lifetime results. With our proposition, advertiser lifetime can be 281 times higher than with the SIG Profile when using ST BlueNRG, reaching a lifetime of 7 years and 4 months with an average advertiser DL of  $\approx 4s$ , which is considered a good user experience for this use case. The SIG Profile provides a lifetime of 9.5 days. This proves that BLE is a suitable technology for this kind of applications when using an optimal configuration. Lifetime on the scanner side improvement is not achieved with the proposed configuration as extending the advertiser lifetime is the priority.

## VII. CONCLUSIONS

We have extended an existing model [4] and proposed a novel parameter optimization method to compute optimal parameters for BLE devices depending on application requirements. This model as well as the parameter optimization are validated experimentally. We show that BLE is suitable for a wide range of IoT applications if parameters are chosen with care. We obtain the appropriate  $T_{SI} - T_{SW} - T_{AI}$  configuration that guarantees a Discovery Latency which satisfies specific requirements for any use case, such as maximum acceptable

Table VIII: Medical telemetry system DL, energy consumption and lifetime results

SIG configurations	TI		BlueNRG		
	Advertiser	Scanner	Advertiser	Scanner	
$T_{SI} = 60ms, T_{SW} = 30ms$ $T_{AI} = 20ms$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	210.27 $\mu$ J 26.3ms 103.1s 3.82 days	3.7mJ 131.9ms 285ms 1.76 days	63.32 $\mu$ J 25.4ms 102.1ms 9.55 days	1.6mJ 129.3ms 282.4ms 2.56 days
$T_{SI} = 60ms, T_{SW} = 30ms$ $T_{AI} = 30ms$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	208.33 $\mu$ J 35.4ms 143.1ms 5.54 days	3.9mJ 140.3ms 337.7ms 1.76 days	62.78 $\mu$ J 34.5ms 142.1ms 13.82 days	1.7mJ 137.7ms 335.1 2.56 days
<b>Proposed this work</b>	<b>Advertiser</b>	<b>Scanner</b>	<b>Advertiser</b>	<b>Scanner</b>	
$T_{SI} = 500ms, T_{SW} = 300ms$ $T_{AI} = 10.24ms$	$E_{avg}$ $DL_{avg}$ $DL_{wc}$ Lifetime	145.43 $\mu$ J 4.1s 10.25s 4.27 years	204mJ 7.03s 21.83ms 1.46 days	53.5 $\mu$ J 4.09s 10.25s 7.36 years	88.2mJ 7.06 21.82s 2.11 days

critical latency. We provide results for DL, energy consumption and battery lifetime for two typical IoT test cases: retail store and medical telemetry. We show that we can obtain huge gains compared to SIG Profiles recommended configurations and provide multi-year lifetime instead of a few days or weeks.

## REFERENCES

- [1] Bluetooth SIG. Specification of the Bluetooth System. 02 December 2014. Core Version 4.2. [Online]. Available: <https://www.bluetooth.com/specifications/bluetooth-core-specification/legacy-specifications>
- [2] S. Kamath and J. Lindh, "Measuring Bluetooth Low Energy Consumption," Texas Instruments, Tech. Rep. Application Note AN092, 2010.
- [3] J. Liu, C. Chen, and Y. Ma, "Modeling neighbor discovery in bluetooth low energy networks," *IEEE communications letters*, vol. 16, no. 9, pp. 1439–1441, 2012.
- [4] P. Kindt, D. Yunge, R. Diemer, and S. Chakraborty, "Precise Energy Modeling for the Bluetooth Low Energy Protocol," *CoRR*, vol. abs/1403.2919, 2014. [Online]. Available: <http://arxiv.org/abs/1403.2919>
- [5] K. Cho, W. Park, M. Hong, G. Park, W. Cho, J. Seo, and K. Han, "Analysis of latency performance of Bluetooth low energy (BLE) networks," *Sensors*, vol. 15, no. 1, pp. 59–78, 2014.
- [6] W. S. Jeon, M. H. Dwijaksara, and D. G. Jeong, "Performance Analysis of Neighbor Discovery Process in Bluetooth Low-Energy Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1865–1871, 2017.
- [7] Bluetooth SIG. Adopted Bluetooth Profiles, Services, Protocols and Transports. 2018. [Online]. Available: <https://www.bluetooth.com/specifications/gatt>
- [8] A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, "Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications," in *IEEE International Conference on Communications*, 2018.
- [9] Bluetooth SIG. Proximity Profile specification. 14 July 2015. Version 1.0.1. [Online]. Available: <https://www.bluetooth.com/specifications/gatt>
- [10] J. Liu and C. Chen, "Energy analysis of neighbor discovery in Bluetooth Low Energy networks," *Nokia(nd)*, 2012.
- [11] STMicroelectronics. BlueNRG Current Consumption Estimation Tool. 2017. Version 1.4. [Online]. Available: <http://www.st.com/en/embedded-software/stsw-bnrg001.html>
- [12] Université Grenoble Alpes, Grenoble INP / UJF. WaIT - Wireless Testbed. 2013-2014. [Online]. Available: <http://wait.forge.imag.fr/>
- [13] R. Knoblauch, M. Pietrucha, and M. Nitzburg, "Field studies of pedestrian walking speed and start-up time," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1538, pp. 27–38, 1996.
- [14] R. W. Bohannon, "Comfortable and maximum walking speed of adults aged 20–79 years: reference values and determinants," *Age and ageing*, vol. 26, no. 1, pp. 15–19, 1997.
- [15] H. Burri and D. Senouf, "Remote monitoring and follow-up of pacemakers and implantable cardioverter defibrillators," *Europace*, vol. 11, no. 6, pp. 701–709, 2009.