Open access • Proceedings Article • DOI:10.1109/ICC.2018.8423011

Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications

Andreina Liendo, Andreina Liendo, Dominique Morche, Roberto Guizzetti ...+1 more authors

Institutions: STMicroelectronics, Commissariat à l'énergie atomique et aux énergies alternatives, Grenoble Institute of Technology

Published on: 20 May 2018 - International Conference on Communications

Topics: Bluetooth, Efficient energy use and Wireless sensor network

Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications

Andreina Liendo, Dominique Morche, Roberto Guizzetti, Franck Rousseau

Efficient Bluetooth Low Energy Operation for Low Duty Cycle Applications

Andreina Liendo*‡¶, Dominique Morche‡, Roberto Guizzetti¶ and Franck Rousseau*
*Grenoble Alps University, Grenoble Institute of Technology, France
Email: firstname.lastname@imag.fr
‡CEA-LETI, MINATEC, 38000 Grenoble, France
Email: firstname.lastname@cea.fr
¶STMicroelectronics, 38920 Crolles, France
Email: firstname.lastname@st.com

***Abstract*—Bluetooth Low Energy (BLE) has shown higher energy efficiency and robustness than other well known Wireless Sensor Network (WSN) protocols, making it a strong candidate for implementation in IoT scenarios. In addition, BLE is in almost every smartphone, turning it into perfect ubiquitous remote controls for smart homes, buildings or cities. In this paper we propose new operating modes for BLE to provide a much wider applicability range covering asynchronous, aperiodic and very low frequency scenarios while extending lifetime, without modification of the existing BLE specification. In order to estimate battery lifetime, we propose a coin cell battery model. We show how to efficiently adapt the protocol for different IoT scenarios based on their requirements, thus extending battery lifetime as much as possible while guaranteeing acceptable latency. Our results on two typical test-cases show that by using an optimized configuration and appropriate operating mode based on scenario requirements, lifetime can be increased up to $\approx 2\times$ (4.1 to 7.3 years) for a BLE master in a first case, and $\approx 58\times$ (3.05 months to 14.82 years) for a BLE slave in a second case. This shows BLE compatibility with a vast amount of IoT-related use cases.**

***Index Terms*—Internet of Things; Bluetooth Low Energy; Energy Consumption; Battery Lifetime; Critical Latency; Wireless Sensor Networks.**

I. Introduction

BLE is a *de facto* standard [1] for low-energy short-range personal communications, and a key radio technology for the IoT [2]. It is one of the most widespread wireless communication technology today. BLE is designed to provide lower energy consumption [3], [4], security, ease of use and to avoid interference with higher power IEEE 802.11 networks. Although it implements a duty cycle, there is a great waste of energy because devices do not exclusively communicate for application data exchange. Devices spend a lot of time waiting for packet arrivals, channel sensing, and receiving/transmitting packets intended for connection maintenance and synchronization activities.

Typical applications are heart-rate monitors sending periodic measurements, wearable monitors tracking user activity [5] and smart watches. The communication pattern is the same for these applications. First, after a trigger from the user, pushing a button or starting an application, the two devices need to detect each other using the *neighbor discovery* (ND) procedure. Then, they establish a connection and once in *connected mode* (CM) they can exchange data. ND is a very energy-hungry non-deterministic process since it relies on one device sending advertising packets alternatively on 3 specific channels, while the other one is asynchronously scanning the same channels to catch one of these packets and proceed with connection establishment. BLE devices remain in connected mode for long periods, transferring small amounts of data on a periodic basis, making ND just a small percentage of the activity compared to the total duration of the devices operation.

The design of BLE restricts the period of communication between two devices in connected mode to a certain maximum to cope with clock drift and maintain tight synchronization. Therefore, for applications with very infrequent and aperiodic communications patterns, a considerable amount of energy is spent in maintaining the synchronization rather than transmitting the application data itself. Energy consumption during ND depends on BLE device configuration parameters [6], whereas energy consumption in CM, is mostly application dependent (eg. activity rate and packet size).

Aiming to make BLE compatible with a much wider range of applications compared to what is typically implemented today, we first analyze BLE communications with respect to typical IoT-related application requirements in Section II. We propose a classification of the different types of existing scenarios according to their requirements and we propose the implementation of application level BLE modes of operation in Section III. This is done without any modification to the existing standard Finally, we demonstrate in Section IV how the different modes of operation fit for each type of scenarios, while ensuring an optimal utilization of battery energy. Results are given in terms of coin cell battery lifetime, for which we propose a battery lifetime estimation model.

Our work builds on a precise Discovery Latency (DL) and energy model [7] from the literature that we have implemented in Matlab. The model has been optimized and experimentally validated [8], in order to give accurate performance results since the model takes into account device behavior at different levels (communication and application level). We also implement the proposed ND parameter optimization method [8] in order to minimize energy consumption for a wider range of IoT use cases, by combining the parameter optimization and

new application level operating modes.

We show that our approach allows to use BLE for long-running applications with low activity rates that have typically been overlooked until now. We consider two typical test-cases, Temperature and Humidity Monitoring, and Light Switch, for which we show improvement in battery lifetime.

II. BLE OVERVIEW AND LIMITATIONS

In order to familiarize the reader with BLE operation, in this section we introduce the BLE communication modes as well as BLE limitations for typical scenarios. However, we refer the reader to [6], [8].

A. BLE overview

The BLE protocol specification establishes five basic states through which at least two devices must transit in order to establish a communication: `standby`, `advertising`, `scanning`, `initiating`, and `connection` (see Figure 1). BLE devices use two different modes of communication: ND and CM. During the former, communication is asynchronous and devices transit through 4 states `standby`, `scanning` and `initiating` states for scanners, or `standby` and `advertising` states for advertisers, using only the 3 advertising channels. In CM a device transits through `connection` and `standby` state, and uses the 37 data channels. For more detail we refer the reader to [6].

According to the Bluetooth specification [6], a BLE scanner periodically scans advertising channels 37, 38 and 39, expecting to receive advertising information from other devices. An advertiser device may send packets to the advertising channels to indicate that it is discoverable or connectable, or to broadcast data.

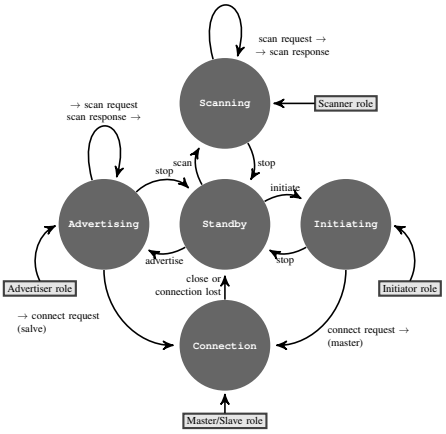

Figure 1. BLE FSM

After successfully receiving an advertising packet, the initiator sends a *connection request* to the advertiser in order to set the timing of the connection. The master has the flexibility to schedule the starting moment of a connection at a time of its choosing called *anchor point*. The slave senses the channel waiting for the first packet from the master during a time duration called *transmit window*. The connection request packet contains three parameters used to determine the *transmit window*: the *connection interval* T_{CI} (which is defined later), the *transmit window offset* which is a multiple of $1.25\,\text{ms}$ in the range of $[0, T_{CI}]$, and the *transmit window size* which is a multiple of $1.25\,\text{ms}$ in the range of $[1.25, \min(10, T_{CI} - 1.25)]$ ms as illustrated in Figure 3. After successfully receiving the first packet from the master the slave will send application data.

B. BLE limitations for typical scenarios

BLE is designed for periodic transfers of very small amounts of data, such as beacons providing proximity and localized information, and leisure or medical devices monitoring vital parameters. One important aspect of energy consumption evaluation for BLE is the suitability of the system behavior to the scenario. This behavior is specified in a collection of profiles associated to the core Bluetooth specification (see the Bluetooth SIG Profiles [9]), where two advertising and scan modes are defined: if a connection is not established in the first $30\,\text{s}$ the devices switch to a lower power mode. However, it has been shown [8], that using the right method, a set of parameters can be chosen based on use case requirements, thus obtaining a much better performance compared to the Bluetooth SIG Profiles. On the other hand, BLE does not seem suitable for long running applications with few events, that nevertheless constitute a large portion of envisioned IoT applications. In this work we consider applications such as metering and home automation that are typical of the Smart Building IoT. Such network comprise tens of nodes that communicate relatively infrequently, every 5 to 15 minutes or only a few times a day, with only a few bytes of payload and relaxed latency requirements of $200\,\text{ms}$ or above [10]. Among some of the specific uses cases within this category we can list: HVAC, lighting control, structural integrity monitoring and access control.

III. LOW DUTY CYCLE APPLICATIONS WITH BLE

Below, we first give a clear problem formulation and then we review the scenarios that we envision to finally explain how we determine proper parameters to achieve very long lifetimes.

A. Problem formulation

Even though BLE has been proved to be more energy efficient and robust than other WSN protocols [3], [4], there is a need for adapting the protocol to more generic IoT scenarios, where traffic patterns are different, with scarcer communication and where applications need to run autonomously during several years. In connected mode, BLE devices have to interact regularly for two main reasons. First, to guarantee a *critical latency* (CL), which is the maximum communication latency that guarantees acceptable operation or user experience, and it is application dependent. Second, to guarantee correct synchronization, which depends on system constraints, mostly on the oscillator accuracy that has a direct impact on clock drift. This second constraint together with the targeted operating conditions, for example the temperature range, define the

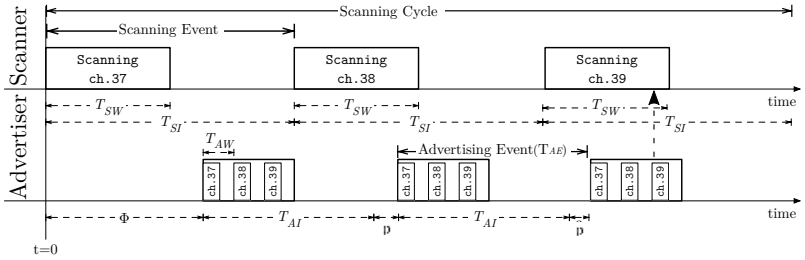

Figure 2. BLE Neighbor Discovery

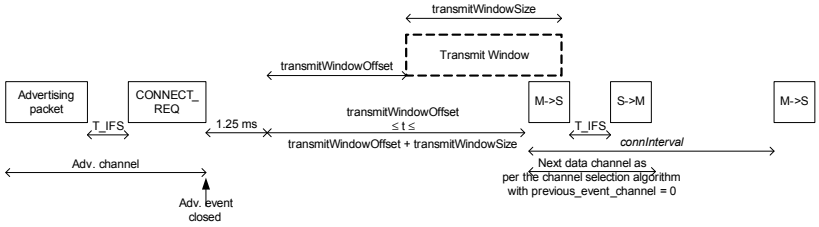

Figure 3. Master's view on connection setup with a non-zero transmit window offset (taken from [6])

maximum communication period for the system, as shown in Figure 4. A BLE module transmits application data during a connection event. The timing of connection events is determined by two parameters: T_{CI} and *Slave latency*. According to the specification T_{CI} has a maximum value of 4 s. Also, the slave latency should not cause a supervision timeout, which means that a slave should at least communicate every 32 s.

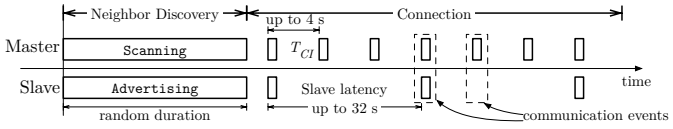

Figure 4. BLE communication pattern

In the case of rare applications events, this causes high communication overhead and represent a huge waste of energy: supposing that even though two devices need to communicate every X min in average, the master would have to wake up every 4 s and the slave every 32 s in order to remain synchronized. For low frequency application events, it seems reasonable to use a communication pattern where devices establish a connection only when needed instead of maintaining a connection, therefore forcing the devices to perform ND regularly. But since ND is a non-deterministic energy hungry process, we need to determine the point where the results are optimal. As it has been shown before [7], [8], *energy consumption* and DL can vary greatly depending on the ND parameters chosen for both scanner and advertiser. Moreover, changing parameters to improve energy consumption on one side will degrade performance on the other side, however a trade off between scanner and advertiser energy consumption can be achieved by finding the right parameters, as we show in Section IV-B3.

B. Scenarios classification

There are two types of events occurrence with respect to their relation to time, *synchronous*, where events timings are known in advance, and *asynchronous*, where events happen randomly. If we now consider communications, we must consider CL, *critical latency*, as well. We present a classification of the scenarios according to their CL and frequency of events occurrence.

1) *Continuous high frequency* are scenarios for which the CL is low, in the order of seconds or less, no matter if events are synchronous or asynchronous, to provide the illusion of real-time, like heart rate monitors that communicate measurements several times per minute to trackers. This is the typical scenario for BLE.
2) *Random low frequency* are scenarios for which the period of interaction is large but events are asynchronous and need a low CL. The light switch scenario falls into this category, switches are used a few times a day, but CL must remain below 200 ms to provide acceptable user experience when turning on or off the light.
3) *Periodic low frequency* are scenarios for which the period of interaction is large and there is a very relaxed CL. We can accommodate asynchronous events, as long as we define an application-level duty cycle, where the application guarantees periodic communication episodes enforcing the CL. This is typically the case for the temperature and humidity monitoring scenario.

C. BLE proposed operating modes

We define the operating modes as different ways of implementing the communication modes of BLE (explained in section II), allowing devices to be compatible with different type of applications while extending battery lifetime. We

propose three possible modes of operation from which to choose depending on the type of scenario:

1) *Classic BLE:* devices perform ND, normally triggered by an action from the user and once they discover each other they establish a connection and stay in CM continuously and for long periods of time. This is the typical mode of operation for BLE.
2) *Fully asynchronous BLE:* the scanner listens asynchronously waiting for a packet from an advertiser. Once the data packet is received the advertiser goes to sleep mode to save energy and the scanner goes back to scanning state, in this case the scanner should be powered by a permanent source of energy. This is the mode where data could be sent within advertising packets for example and has not been previously evaluated for autonomous and long-running applications.
3) *Duty cycled BLE (DC-BLE):* application data exchange is quite rare, so both devices go to sleep mode for long periods of time, but they must perform ND prior to a data exchange, so optimization of this phase is necessary in order to achieve better energy consumption compared to classic BLE. We propose to enforce a duty cycle on top of BLE (our main contribution).

Figure 5 shows the typical behavior of single master to single slave using the different modes of operation. Detailed aspects of the DC-BLE mode are shown in Figure 6, where we show the least favorable case, when application data must be sent after setting up a connection. If the size of the data is small enough it can be optimized, using the non-connectable undirected advertising up to 31 B of data, or using the scannable undirected advertising for more than 31 B by sending application data within the scan response packets, so a connection establishment is not necessary in that case.

For the DC-BLE, we apply the concept of window widening as defined in the BLE specifications, not only at communication level but also at application level. Node oscillators do not operate at the exact same frequency. This brings uncertainty in the slave of the exact timing of the master's anchor point due to clock drift. During CM, slaves are required to re-synchronize to the master's anchor point at each connection event where it listens for the master. BLE spec establishes a listening time called `windowWidening` which is the time before and after the `transmit Window` that the slave must listen to ensure a synchronization with the master. Analogously, for DC-BLE mode, where we set up a duty cycle on top of classic BLE, clock drift must be taken into account from the scanner side. During each communication event, we consider a time that the scanner should listen before and after the scanning event in order to perform the ND at approximately the same time for both advertiser and scanner, thus increasing the probability of obtaining a low DL. We call this wwDC, more details can be seen in Figure 6. We use the equation given in the specification for CM as shown below:

$$\texttt{wwDC} = \frac{\texttt{masterSCA} + \texttt{slaveSCA}}{10^6} \times \texttt{tSLA} \qquad (1)$$

Where `masterSCA` and `slaveSCA` are the master and slave sleep clock accuracy and `tSLA` is the time since the beginning of the last scanning event. We used a 40 ppm clock accuracy as it is a typical value for WSN nodes. Then we use the obtained window widening value to determine the minimum time devices should spend in ND before being able to exchange data. We use this value as the CL for the parameter optimization method from [8] in order to determine the rigth set of $T_{SI} - T_{SW} - T_{AI}$ parameters.

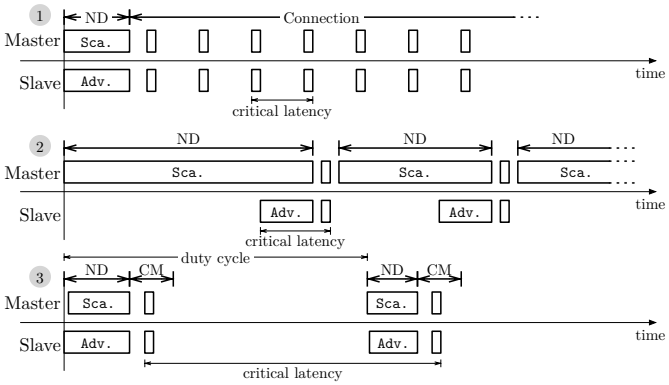

Figure 5. Proposed operation modes

D. Proposed test-cases

Temperature and Humidity Monitoring: let's consider one node communicating to a central device to send temperature and humidity readings. The sensing node would periodically wake up, collect the environmental data and transmit this data. Temperature and humidity are monitored at several moments in the day, we consider that the average timing between transmissions is 43 minutes with no constraint on the latency [11] so th main constraint is energy consumption.

Light Switches: we also consider a use case in which one or several light switches are used to control one or several lamps. In this case, the main constraint is a CL of 200 ms, which is the maximum latency for a good user experience [10]. Typically, switches are operated only a few times a day (e.g. office and home environment).

E. Assumptions and limitations

In the following, we take the classic assumptions found in the literature with which we compare to. We focus primarily on point-to-point master-slave communication that are typical for BLE, which is asymmetric in nature. However, we give some insight on how BLE behaves in the more general scenario of multi-hop WSN, which is symmetric and where nodes must act as masters as well as slaves. For the estimation of the energy consumption we do not take into account security, so no Message Integrity Check is considered, which saves 4 B. Also, we consider that no user interaction is required, that $\texttt{transmitWindowOffset} = 0$, and that channel conditions are ideal with no interference or collisions.

Multiple slaves cause problems when using connected mode: the master has to schedule connection events for all

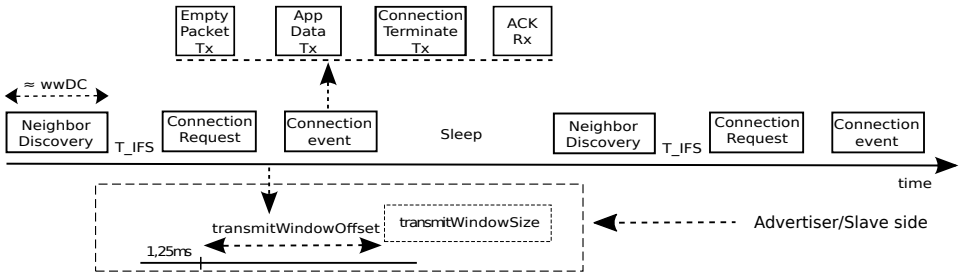

Figure 6. Proposed DC-BLE from Master's perspective

of them while avoiding collisions and keeping some time for scanning for additional devices. In fully asynchronous and DC-BLE, we do not keep master and slave connected, hence we do not face this problem. Since we consider only one master with one slave in our evaluation, this is the best situation for classic BLE with respect to our proposition.

IV. Evaluation and results

In this section we first introduce our proposed battery lifetime estimation model, then we present the lifetime results for the two IoT test-cases and finally we explain how our proposed operating modes compare to classical operating mode of BLE and we establish use case conditions for which one operating mode or another is preferable.

1) Battery Lifetime: The main battery characteristic is its rated capacity which varies with the discharge rate. The total battery life can be estimated from Eq. 4 (Figure 7), so the lower the discharge rate, the longer the battery lifetime. In order to compare the lifetime of the nodes for all the communication types, we use the parameters of a Panasonic CR2032 coin cell battery type with a nominal capacity of 225 mAh for a continuous load of 0.2 mA. We take into account how the battery capacity is affected by the usage pattern, as the capacity found in a battery data-sheet is not fixed for all conditions.

A wireless application subjects the battery to different conditions, the radio circuitry considered in this paper can draw anywhere from $1\,\mu$A in sleep mode to 17.5 mA during Rx/Tx, which far exceeds the rated drain current condition, for which the battery capacity is given in the battery data-sheet. The impact of this variations for a typical coin cell battery supporting typical WSN applications has been evaluated in previous work [12], from which we can assume that, as long as the highest current drain is less than 30 mA and the average current is less than the rated drain current specified on the data-sheet, the battery capacity will remain close to the nominal value, otherwise the battery capacity must be recalculated.

We recalculate the battery capacity for average current higher than the nominal value based on Peukert's law as shown in Figure 7. According to the curve for Capacity *vs.* Load Resistance found in the data sheet, we estimate a Peukert's constant of 1.1 and we use Eq. 2 or 3, where I_r is the nominal discharge rate, I is the average current consumption estimated for the different scenarios, C_r is the rated battery capacity and H is the rated discharge time estimated based on the nominal

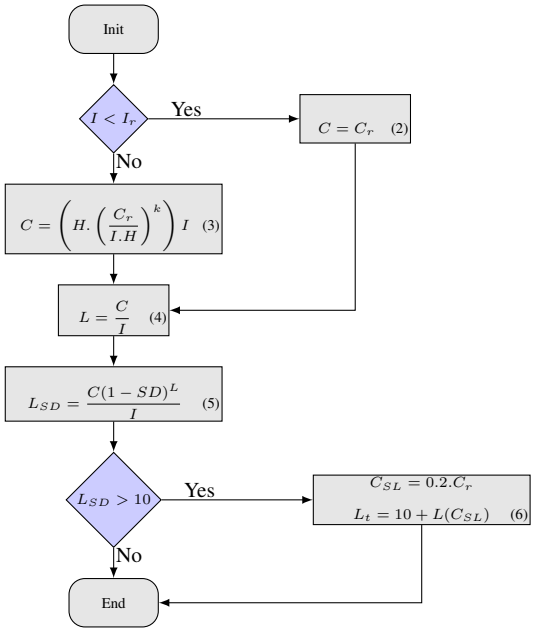

Figure 7. Battery lifetime estimation

values and which is 1125 hours. Lifetime is first estimated according to Eq. 4, then we consider 1% of self discharge (SD) per year as shown in Eq. 5. In addition, a real maximum lifetime for typical WSN applications of $\approx$15 years has been estimated in previous studies [13], based on this we assume no more than 20% of battery capacity C_{SL} is left after 10 years of battery usage due to the consumption pattern (I) seen in our simulations, so the total battery lifetime L_t is given by Eq. 6.

A. Lifetime results on test-cases

In this section we present the lifetime results when implementing the different operating modes of BLE for the two chosen test-cases explained in Section III-D. First we consider one single node communicating to a central device to send temperature and humidity readings every 43 minutes in average [11]. We compare this scenario using DC-BLE against classic BLE and found that for TI device the advertiser/slave lifetime can reach more than 14 years, approximately $1.13\times$ compared to classic BLE, whereas the scanner/master lifetime can reach more than 7 years when using the DC-BLE scheme which is

$\approx 1.8\times$ the lifetime of the master implementing classic BLE, see Table I. When using classic BLE, ST master device is $2.7\times$ better than TI master because the Rx/Tx current consumption of the TI device is $2.3\times$ greater than that of ST. On the other hand, the performance of ST master device in DC-BLE mode is not better than in classic mode, due to the clock drift, which causes a large waste of energy. This could be improved by reducing the current consumption of the chip during the sleep mode (which is 1.15μA).

For the light switch control system, we first consider one single switch with a BLE device powered with a coin cell battery that communicates to a bulb with a BLE device that can be connected to the mains. The scanner mode is configured at the bulb end and as it is connected to a permanent source of energy, it is feasible to implement continuous scanning during ND, thus ensuring the minimum discovery latency and energy consumption on the advertiser side. As the CL is required to be 200 ms [10], when implemented with classic BLE, the minimum T_{SI} is set to 200 ms which would lead to a lifetime of approximately 3.05 months (0.25 year), whereas when communicating asynchronously only 20 times per day sending a few bytes of application data, the advertiser lifetime could be extended up to 14.82 years ($58.3\times$) in the worst case where the data is not transmitted in the advertising packet but a connection is established and terminated right after the application data exchange as shown in Figure 6. For further details see Table I.

Table I
LIFETIME IN YEARS FOR TWO TEST-CASES, WITH TI CC2540 AND ST BLUENRG DEVICES, IN SCANNER/MASTER AND ADVERTISER/SLAVE ROLES

		Temperature and Humidity Monitoring		Light Switches	
		TI	ST	TI	ST
Classic BLE	S/M	4.1	10.88	0.25	1.1
	A/S	13.05	14.08	0.25	0.76
Fully Async	S/M	N/A	N/A	0.25	0.25
	A/S	N/A	N/A	14.82	14.26
				58.3×	18.72×
DC-BLE	S/M	7.3	10.46	N/A	N/A
		1.8×	.96×		
	A/S	14.78	14.25	N/A	N/A
		1.13×	1.01×		

B. General Results

1) Advertiser energy consumption: When using DC-BLE, regardless of the chosen parameters and even though ND is performed each time prior to a data exchange, our results show that energy consumption at the advertiser/slave side will always be better than classic BLE for periodic low frequency scenarios as long as the communication period (time between each application data exchange) is greater than 76 seconds for TI device and 260 second for ST device as shown on Figure 8 and 9. In other words, DC-BLE can be implemented for applications where CL can be relaxed up to 76 s/260 s or above while still providing a good user experience. On the other hand, for random low frequency applications, the fully asynchronous BLE is always better than classic BLE at the advertiser/slave side as communication is carried out only when needed and using the minimum amount of energy possible.

2) Scanner energy consumption: We estimate the lifetime of the scanner when using DC-BLE and compare this results against classic BLE. We find that the former provides longer lifetime for periodic and low frequency scenarios, as long as the CL is greater than 86 s and is never better for ST device because its performance is already better using classic BLE. The only way the ST master could surpass classic BLE using DC-BLE would be by reducing the current consumption during sleep mode. It is important to highlight that at this point (when master lifetime is greater using DC-BLE), the lifetime of the DC-BLE slave is not better than the slave using classic BLE and as it can be seen in Figure 8, communication is asymmetric, where the advertiser lifetime reaches up to twice the scanner lifetime. Further discussion about energy symmetry can be found in IV-B3.

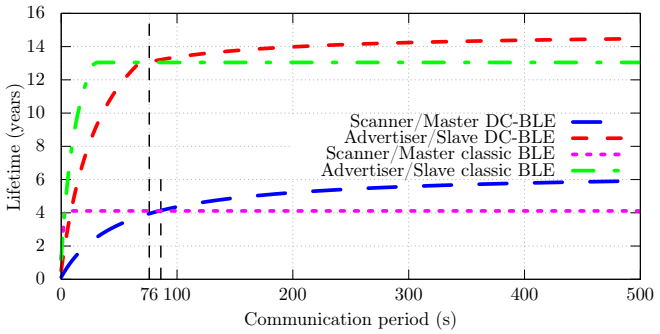

Figure 8. Advertiser/Slave and Scanner/Master Lifetime when using TI C2540 devices

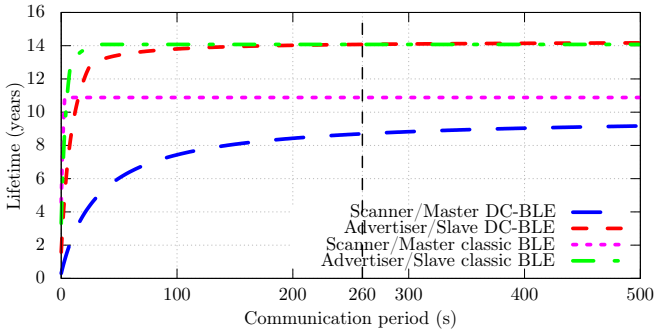

Figure 9. Advertiser/Slave and Scanner/Master Lifetime when using BlueNRG devices from ST Microelectronics

For random low frequency applications, fully asynchronous mode is always worst than the classic mode at the scanner side. A permanent source of energy is the only choice since when powered on a coin cell battery, the lifetime would be reduced to only about 15 h. The main focus when using this

mode is to increase the advertiser lifetime. Additionally this configuration can support many advertisers/slaves at the same time while having the minimum latency and energy consumption since these devices do not need to be associated to the master. An example could be the light switch scenario where several switches can control one or many bulbs completely asynchronously, where the role of the switch can also be a smartphone with no need to trigger an application on the scanner side.

3) Scanner/Advertiser Consumption Trade-off: Mesh networks tend to break the master-slave assumption on which Bluetooth is built. Recent version 5.0, allows many-to-many connection by adding secondary advertising packets which are transmitted *connectionless* using the data channels, allowing to broadcast synchronous data [9]. Up to version 4.2, the master can connect to several slaves in the same network while a slave can only connect to one single master. However, multi-hop BLE communication is feasible with only advertising packets, and in this case it is desirable that scanners and advertisers have a equal lifetimes. We consider random low frequency scenarios, where nodes should provide some kind of symmetry in energy consumption and be able to relay bidirectional traffic. Let's consider the fully asynchronous mode, where the master does continuous scanning allowing to scan other devices and achieving the best performance on the slave side, but depleting the scanner's battery very rapidly. A lifetime trade-off between master and slave can be achieved by decreasing the duty cycle on the scanner side, thus extending its lifetime at the expense of decreasing the slave's lifetime and increasing the DL. As shown in Figure 10, symmetric lifetime is possible when scanner duty cycle is 0.24% for the TI device and 0.58% for the ST device while the communication period on the slave side is of $1.06\,\text{s}$ and $1.1\,\text{s}$ respectively. As a drawback, DL will not be less than $1.5\,\text{s}$ in average, making this solution not compatible for applications with lower required CL.

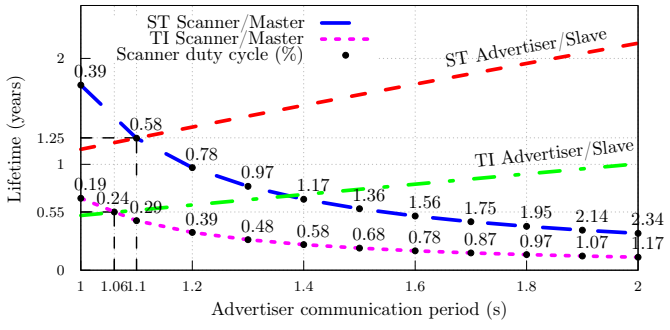

Figure 10. Master (central) and Slave (peripheral) trade-off for fully asynchronous mode of operation. Master's lifetime depending on the duty cycle when implementing $T_{SI} = 10.24\,\text{s}$ for the TI device and $T_{SI} = 5.12\,\text{s}$ for the ST device. Slave's lifetime depending on the communication period (application dependent). ST device performance when symmetry is achieved, is $\approx 2.3\times$ better than TI device performance.

V. Conclusion

We have defined three types of operation mode: classic BLE, fully asynchronous BLE and DC-BLE without modification of the specification. The idea is to implement a given scheme depending on application requirements, for which we have categorized the applications in three types: continuous high frequency, random low frequency and periodic low frequency. We have seen that for periodic low frequency applications, performance is better in terms of energy consumption when using DC-BLE if the critical latency is greater than 76 s, which covers a wide range of applications. Otherwise it is better to use classic BLE. The fully asynchronous BLE mode dramatically extends battery lifetime of slave devices, but a permanent source of energy is required on the scanner/master's side. However a trade off can be obtained by decreasing scanner duty cycle at the expense of increasing energy consumption at the slave side in order to reduce energy consumption on the scanner/master's side. We have presented the results for two devices from different manufacturers and shown that the results may vary significantly depending on the consumption of internal radio states.

References

[1] M. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, "Internet of Things in the 5G Era: Enablers, Architecture and Business Models," *Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–17, 2016.

[2] K.-H. Chang, "Bluetooth: A Viable Solution for IoT?" *IEEE Wireless Communications*, vol. 21, pp. 6–7, 2014.

[3] A. Dementyev, S. Hodges, S. Taylor, and J. Smith, "Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Node in a Cyclic Sleep Scenario," in *IEEE International Wireless Symposium (IWS)*, 2013, pp. 1–4.

[4] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4," in *Wireless Communications and Networking Conference Workshops (WCNCW)*, 2012, pp. 232–237.

[5] C.-C. Yang and Y.-L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *MDPI Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.

[6] 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

[7] 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

[8] A. Liendo, D. Morche, R. Guizzetti, and F. Rousseau, "BLE Parameter Optimization for IoT Applications," in *IEEE International Conference on Communications*, 2018.

[9] Bluetooth SIG. Adopted Bluetooth Profiles, Services, Protocols and Transports. 2018. [Online]. Available: https://www.bluetooth.com/specifications/gatt

[10] T. Jorgensen and N. T. Johansen, "Z-wave as Home Control RF Platform," *Zensys A/S*, 2005.

[11] X. Zhang and B. Chang, "Research of temperature and humidity monitoring system based on WSN and fuzzy control," in *IEEE International Conference on Electronics and Optoelectronics (ICEOE)*, vol. 4, 2011.

[12] K. Furset and P. Hoffman, "High pulse drain impact on CR2032 coin cell battery capacity," Nordic Semiconductor and Energizer, Tech. Rep., 2011, technical Report. [Online]. Available: http://www.eetimes.com/ContentEETimes/Documents/Schweber/C0924/C0924post.pdf

[13] M. Tomasz. Long live coin cells: Getting years of battery life from small batteries in low power devices. 18 September 2013. [Online]. Available: http://www.edn.com/design/power-management/4421334/2/Long-Live-Coin-Cells--Getting-years-of-battery-\\life-from-small-batteries-in-low-power-devices