

Article

Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimplicTI

Konstantin Mikhaylov ^{1,*}, Nikolaos Plevritakis ² and Jouni Tervonen ¹

¹ Oulu Southern Institute, University of Oulu, Ylivieska, 84100, Finland;
E-Mail: jouni.tervonen@oulu.fi

² Department of Electronic Engineering, Technological Educational Institute of Crete, Chania, 73135, Greece; E-Mail: nplevis@chania.teicrete.gr

* Author to whom correspondence should be addressed; E-Mail: konstantin.mikhaylov@oulu.fi;
Tel.: +358-468-841-370.

Received: 17 July 2013; in revised form: 12 August 2013 / Accepted: 14 August 2013 /

Published: 22 August 2013

Abstract: Bluetooth Low Energy (BLE) is a recently developed energy-efficient short-range wireless communication protocol. In this paper, we discuss and compare the maximum peer-to-peer throughput, the minimum frame turnaround time, and the energy consumption for three protocols, namely BLE, IEEE 802.15.4 and SimplicTI. The specifics and the main contributions are the results both of the theoretical analysis and of the empirical measurements, which were executed using the commercially available hardware transceivers and software stacks. The presented results reveal the protocols' capabilities and enable one to estimate the feasibility of using these technologies for particular applications. Based on the presented results, we draw conclusions regarding the feasibility and the most suitable application scenarios of the BLE technology.

Keywords: Bluetooth Low Energy; BLE; 802.15.4, SimplicTI, performance; throughput; latency; efficiency; measurements; energy

1. Introduction

During recent years, energy-efficient short-range wireless communication technologies have become a hot topic for research and development. The efforts of researchers and engineers have increased the energy efficiency of and reduced the monetary costs for wireless data transmission.

Therefore, for many applications today, wireless data transfer appears to be more efficient than data transfer using wired media [1,2].

Nowadays, numerous transceivers implementing the different wireless communication protocols are available on the market. One of the recently suggested protocols is Bluetooth Low-Energy (BLE), which is aimed at applications and products requiring low current consumption and low implementation complexity and having low production costs [3]. During the development of BLE and shortly after its introduction, it was predicted that the protocol would have a very wide application area. For example, in [4] the authors predicted that BLE-based devices would dominate the Wireless Sensor Network (WSN) application market by 2015. Nonetheless, even today, *i.e.*, more than two years after the finalization of the BLE specification and over a year since the appearance of the first commercial BLE transceivers, the features of the protocol itself and the capabilities of the hardware (HW) and software (SW) implementing the protocol are still not broadly known. Therefore, in this paper we study BLE and compare it with a proprietary radio protocol and with IEEE 802.15.4, which is today one of the most popular technologies for energy-efficient short-range wireless data transmission.

The specifics and the major contributions of this paper are the results of the heuristic analysis of the protocols' capabilities and the results of the empirical measurements that we performed using the real-life off-the-shelf transceivers. These data reveal the maximum throughput and the minimum turnaround time one can potentially achieve using the protocols under discussion, and they reveal the values of these parameters for real transceivers. Additionally, in the paper, we discuss the energy consumption of the real-life transceivers implementing the protocols. All the transceivers that we used for our experiments have the same processing core, which is based on the 8051 microcontroller. This enabled us to estimate the complicity of the protocols based on the resources consumed by each protocol stack. These data are presented and discussed in the paper as well.

We first discuss some of the previous research focused on the protocols under discussion in Section 2. In Section 3 we provide a brief overview of the protocols and present the analytic estimations of the maximum throughput and the minimum turnaround time. Section 4 describes the details of our testbed and the experiments. Section 5 presents and discusses in detail the obtained analytic and experimental results. Finally, Section 6 concludes the paper, summarizes the obtained results and discusses the feasibility and the most suitable application scenarios for BLE.

2. Related Work

2.1. IEEE 802.15.4

The initial revision of the IEEE 802.15.4 standard was introduced in 2003. During the following years, multiple research papers discussing different aspects of the IEEE 802.15.4 protocol were published.

The performance of IEEE 802.15.4 has been studied in [5]. There, the authors state that for the single-hop scenario, the upper bound of the throughput in the IEEE 802.15.4 nonbeacon-enabled network is defined by the time required to transfer the frame header, data, acknowledgement (if used) and wait period between frames. According to their analysis, the maximum effective throughput for unacknowledged single-hop data transmission is 140.9 kbit/s [5]. Nonetheless, the authors did not account

for the fact that some service operations (including, e.g., the carrier sense multiple access (CSMA) algorithm) can be executed during the interframe space (IFS) period [6].

In [7] and [8], the authors accounted for the possibility of executing the CSMA algorithm during the IFS. In [7], Latre *et al.* used the default values for the variable defining the operation of the CSMA algorithm and obtained a maximum throughput of 148.8 kbit/s and 162.2 kbit/s for acknowledged and unacknowledged single-hop transmission, respectively. Choi and Zhou [8] optimized the CSMA algorithm parameters and reported maximum single-hop throughput values of 167.6 kbit/s and 189.5 kbit/s for acknowledged and unacknowledged transmission, respectively. In [6] we have confirmed these results and provided the analytic framework for IEEE 802.15.4 throughput analysis for different operation modes. The transmission delay, end-to-end latency and packet delivery rate analyses for the Electrocardiogram (ECG) monitoring system based on IEEE 802.15.4 have been studied by Liang and Balasingham in [9].

2.2. BLE

BLE was introduced as a part of the Bluetooth Core Specification version 4.0 [3] in June 2010. Although quite significant time has passed since the standard was developed, there are only a few research papers discussing BLE.

The implementation of the BLE transceiver or of particular transceiver components was discussed by the authors in [10–12]. In their work, Zhang *et al.* [10] and Masuch and Delgado-Restituto [11] suggested ways to implement the demodulators of the BLE receiver. In [12] Wong *et al.* presented a new low power transceiver chip that supports three different protocols, including BLE.

The feasibility of BLE for real-life applications and the lessons learned while developing applications using BLE technology were discussed in [13–15]. The authors of all these papers considered utilizing BLE for different biomedical use cases.

In [16] the authors analysed the network discovery process in BLE and estimated the average latency and the average energy consumption for it. In [17] and [18] the authors analysed and used the simulation tools to estimate the throughput and the latency for BLE communication. They reported that the maximum application layer throughput for BLE equals 236.7 kbit/s [17]. Additionally, the authors in [17] reported the experimental energy consumption measurements for a real BLE transceiver and pointed out that due to the limitations of the used BLE stack in practice, they were able to obtain a maximum throughput of only 58.48 kbit/s. The energy consumption of BLE was also discussed in [19] and [20]. In [19] the results revealing the Texas Instruments (TI) CC2540 BLE transceiver's power consumption were presented and discussed. In [20] the authors reported and compared the results of the power consumption measurements for the BLE and ZigBee [21] transceivers, showing that BLE's energy utility is 2.5 times better than the one of ZigBee [20].

2.3. SimpliciTI

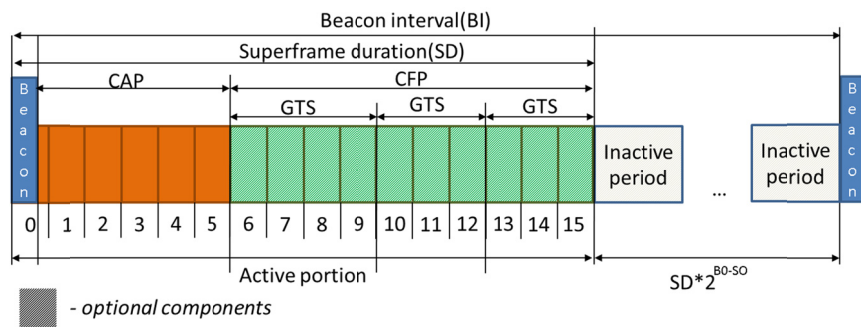
SimpliciTI is a proprietary radio protocol developed by TI for its radio modules, and we are not aware of any research focusing on the performance evaluation of the protocol. Nonetheless, the protocol has been used to implement radio communication in [22–24].

3. Description of the Protocols

3.1. IEEE 802.15.4

The first version of the IEEE 802.15.4 standard was introduced in 2003 under the name IEEE 802.15.4-2003 [25]. Two revisions, namely IEEE 802.15.4-2006 [26] and IEEE 802.15.4-2011 [27], followed in 2006 and 2011, respectively [6]. The standard [27] defines the physical (PHY) layer and the medium access control (MAC) sublayer and aims to provide low complexity, low power consumption and low data rate wireless connectivity among inexpensive devices. The most recent revision of the standard [27] defines twelve PHY options. Of these, the 2450 direct-sequence spread spectrum (DSSS) option, which utilizes the license-free industrial, scientific and medical (ISM) 2.4 GHz band, is the most widely used today [28]. Thus, while speaking about IEEE 802.15.4, we will refer to this PHY layer, unless stated otherwise.

Figure 1. Structure of IEEE 802.15.4 SF (example of a superframe (SF) with six slots for contention access period (CAP) and ten slots for an contention-free period (CFP) with three guaranteed time slots (GTS)) [27].



The standard [27] supports two types of personal area networks (PANs). The first supported PAN type is the beacon-enabled PAN, which contains a coordinator node that periodically transmits beacon frames. The beacon frames are used to synchronize all devices within a PAN and bound the superframes (SFs). A beacon frame contains the data that identify the PAN and describe the used SF structure and can include some user-defined data [27]. As revealed in Figure 1, an SF can have active and inactive portions. During the inactive period, no data transfer is expected and the nodes can switch to a low-power sleep mode to save energy. The active period is divided into 16 equal slots that form a contention access period (CAP) and, optionally, a contention-free period (CFP). The beacon interval (BI) and the SF duration (SD) are defined in the standard by:

$$BI = aBaseSuperframeDuration \cdot 2^{BO} \quad (1)$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO} \quad (2)$$

where SO and BO stand for the SF and the beacon orders, and $aBaseSuperframeDuration$ is a constant equal to 960 symbol periods (*i.e.*, 15.36 ms for 2450 DSSS PHY) and $0 \leq SO \leq BO \leq 14$. As easy to see, the BI and SD for IEEE 802.15.4 2450 DSSS can have the values ranging from 15.36 ms to 251.66 s. The CAP starts immediately after the beacon and should last at least $aMinCAPLength$ symbols (*i.e.*, 440, as defined in [27]) under normal conditions. Therefore, the minimum length of the CAP period for the 2450

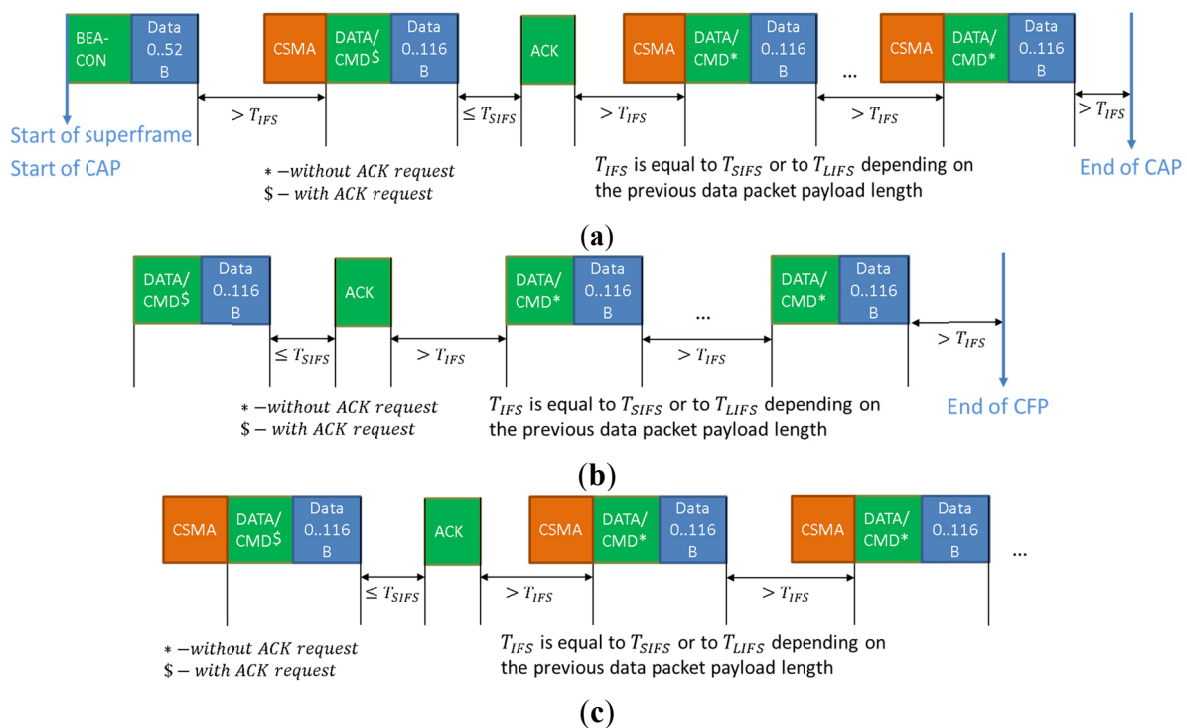
DSSS PHY layer is 7 ms. If used, the CFP starts on the slot boundary immediately following the CAP and closes before the end of the active portion of the SF [27]. The CFP can include up to seven guaranteed time slots (GTSs) that are assigned by the network coordinator and used for unidirectional data transferring between the PAN coordinator and a device associated with the coordinator [27]. A GTS can occupy more than one slot period [27] (see Figure 1).

The second type of network supported by IEEE 802.15.4 is the nonbeacon-enabled one. Although the beacon frames in a nonbeacon-enabled network can be used e.g., to support network discovery, the SFs in those networks are not used.

IEEE 802.15.4-2011 defines two types of channel access mechanisms to be used for different networks [27]. The nonbeacon-enabled PANs use the unslotted Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) algorithm. The beacon-enabled PANs use the slotted CSMA-CA mechanism for transmitting the data and command frames within the CAP. The CSMA-CA is not used for transmitting the beacons and acknowledgement (ACK) frames and for transmission in CFP.

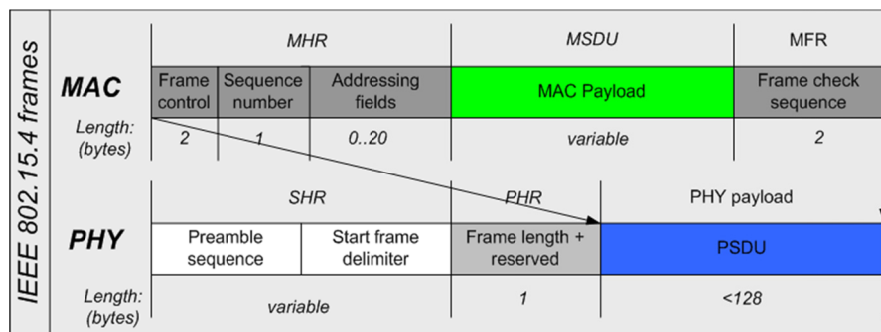
To give a receiver sufficient time for received frame processing, the standard [27] prescribes two successive frames transmitted by a device to be separated by at least an IFS. The length of the IFS after the frame transmission depends on the size of the frame. E.g. for a 2450 DSSS the frame with a MAC protocol data unit (MPDU) of less than 19 bytes should be followed by a short IFS (SIFS), which is equal to 12 symbols [27]. Longer frames must be followed by a long IFS (LIFS) equal to 40 symbols. In the case of acknowledged frame transmission, the IFS starts after the reception of the ACK frame [27]. The timings for data transfer in CAP and CFP of a beacon-enabled PAN and in a nonbeacon-enabled PAN are illustrated in Figure 2a–c, respectively.

Figure 2. (a) Timing for data transfer in CAP in the beacon-enabled IEEE 802.15.4 PAN; (b) Timing for data transfer in CFP in the beacon-enabled IEEE 802.15.4 PAN; (c) Timing for data transfer in the nonbeacon-enabled IEEE 802.15.4 PAN.

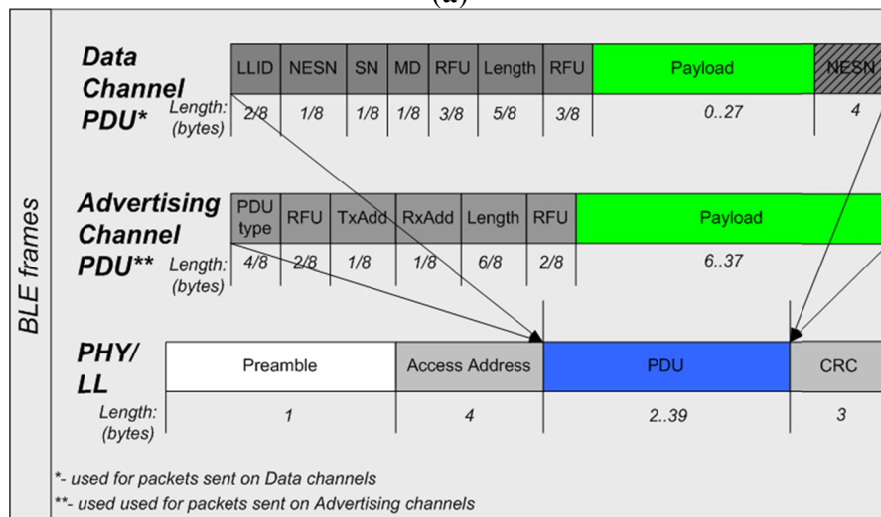


IEEE 802.15.4 [27] defines four possible types of frames: beacon, data, ACK and MAC commands. The general format of frames prescribed by IEEE 802.15.4 for 2450 DSSS PHY is illustrated in Figure 3a. Each frame starts with a synchronization header (SHR) that is used for bit- and byte-wise synchronization. A PHY header (PHR) is used to specify the length of the PHY payload (*i.e.*, MPDU). A MAC header (MHR) stores the required information about the frame type, format of different fields and other features and the sequence identifier of the frame. Additionally, the MHR contains the addressing data, which depend on the type of the frame [27]. E.g., the ACK frames do not have any address data. The length of the addressing data field for a data frame can be between 4 and 20 bytes. The maximum size of the MAC payload (MAC service data unit—MSDU) varies for different frame types. For the data frames, the maximum length of MSDU is 116 bytes. The last two bytes contain the 16-bit ITU-T cyclic redundancy check (CRC) that is calculated over the MHR and MAC payload fields [27].

Figure 3. (a) Data frame format for IEEE 802.15.4; (b) Data and advertising frame formats for BLE; (c) Data frame format for Texas Instrument’s SimpliciiTI.



(a)



(b)



(c)

3.1.1. Maximum Single-Hop Throughput

In [6] we showed that the maximum throughput for the MAC payload (in kbit/s) in the nonbeacon-enabled IEEE 802.15.4 2450 DSSS PAN for acknowledged and unacknowledged single-hop data transfer can be calculated by Equations (3) and (4), respectively. In Equations (3), (4) and onwards n denotes the number of MAC payload bytes in a single frame, τ stands for the radio signal propagation delay, $T_{TXprep}(n)$ and $T_{RXproc}(n)$ account for the time required to process the n -byte data frame before the transmission and after the reception, respectively. The notation $\max(a; b)$ denotes the function that returns a if $a \geq b$ and b if $a < b$. For further details and explanations about the frame format please refer to [6].

$$Throughput_{NBE_ACK}(n, T_{TXprep}, T_{RXproc}, \tau) = \begin{cases} \frac{n \cdot 8}{\max(\tau + 1.056ms + T_{TXprep}(n); T_{RXproc}(n)) + \tau + \frac{n \cdot 8}{250} + 0.544ms}, & n < 8 \text{ bytes} \\ \frac{n \cdot 8}{\max(\tau + 0.544ms + \max(0.64ms; T_{TXprep}(n) + 0.512ms); T_{RXproc}(n)) + \tau + \frac{n \cdot 8}{250} + 0.544ms}, & n \geq 8 \text{ bytes} \end{cases} \quad (3)$$

$$Throughput_{NBE_NACK}(n, T_{TXprep}, T_{RXproc}, \tau) = \begin{cases} \frac{n \cdot 8}{\max(T_{TXprep}(n) + 0.512ms; \tau + T_{RXproc}(n)) + \frac{n \cdot 8}{250} + 0.544ms}, & n < 8 \text{ bytes} \\ \frac{n \cdot 8}{\max(0.64ms; T_{TXprep}(n) + 0.512ms; \tau + T_{RXproc}(n)) + \frac{n \cdot 8}{250} + 0.544ms}, & n \geq 8 \text{ bytes} \end{cases} \quad (4)$$

In a beacon-enabled PAN, the maximum throughput depends on the number of time slots in the CAP (*i.e.*, N_{CAP}) and CFP and can be calculated by Equation (5) using Equations (6) and (7) for acknowledged data transfer or Equations (8) and (9) for unacknowledged transfer. In Equations (6) and (8) $\lceil x \rceil$ denotes the function that rounds x to the nearest integer greater than or equal to x .

$$Throughput_{BE}(n, SD, N_{CAP}, T_{Beacon}, T_{TXprep}, T_{RXproc}, \tau) = Throughput_{BE_CAP}(n, T_{TXprep}, T_{RXproc}, \tau) \cdot \frac{SD \cdot \frac{N_{CAP}}{16} - T_{Beacon}}{SD} + Throughput_{BE_CFP}(n, T_{TXprep}, T_{RXproc}, \tau) \cdot \frac{16 - N_{CAP}}{16} \quad (5)$$

$$Throughput_{BE_ACK_CAP}(n, T_{TXprep}, T_{RXproc}, \tau) = \frac{n \cdot 8}{\lceil (\max(\tau + 0.544ms + \max(T_{TXprep}(n) + 0.832ms; T_{RXproc}(n))) + \tau + 0.544ms + \frac{n \cdot 8}{250}) / 0.32ms \rceil \cdot 0.32ms} \quad (6)$$

$$Throughput_{BE_ACK_CFP}(n, T_{TXprep}, T_{RXproc}, \tau) = \begin{cases} \frac{n \cdot 8}{\max(\tau + 0.736ms + T_{TXprep}(n); T_{RXproc}(n)) + \frac{n \cdot 8}{250} + 0.544ms + \tau}, & n < 8 \text{ bytes} \\ \frac{n \cdot 8}{\max(\tau + 0.544ms + \max(T_{TXprep}(n) + 0.192ms; 0.64ms); T_{RXproc}(n)) + \frac{n \cdot 8}{250} + 0.544ms + \tau}, & n \geq 8 \text{ bytes} \end{cases} \quad (7)$$

$$Throughput_{BE_NACK_CAP}(n, T_{TXprep}, T_{RXproc}, \tau) = \frac{n \cdot 8}{\lceil (\max(T_{TXprep}(n) + 0.832ms; T_{RXproc}(n) + \tau) + 0.544ms + \frac{n \cdot 8}{250}) / 0.32ms \rceil \cdot 0.32ms} \quad (8)$$

$$Throughput_{BE_NACK_CFP}(n, T_{TXprep}, T_{RXproc}, \tau) = \begin{cases} \frac{n \cdot 8}{\max(T_{TXprep} + 0.192ms; T_{RXproc} + \tau) + \frac{n \cdot 8}{250} + 0.544ms}, & n < 8 \text{ bytes} \\ \frac{n \cdot 8}{\max(T_{TXprep} + 0.192ms; 0.64ms; T_{RXproc} + \tau) + \frac{n \cdot 8}{250} + 0.544ms}, & n \geq 8 \text{ bytes} \end{cases} \quad (9)$$

3.1.2. Minimum Single-Hop Turnaround Time

By minimum turnaround time we understand the minimum time required for sending a data frame from node A to node B and for getting a data frame from node B to node A in reply. If the payload of the forward frame (*i.e.*, a frame sent from A to B) is n bytes and the payload of reply frame is m bytes, the minimum turnaround time for acknowledged and unacknowledged transmissions in the nonbeacon-enabled IEEE 802.15.4 PAN is given by:

$$T_{turnaround_{NBE_ACK}}(n, m, T_{RPLgen}, T_{TXprep}, T_{RXproc}, \tau) = 2 \cdot T_{CCA} + T_{Datapkt}(n) + T_{Datapkt}(m) + T_{ACK} = 2.656 \text{ ms} + \frac{(n+m) \cdot 8}{250} \text{ ms} + 2 \cdot (T_{TXprep} + T_{RXproc} + \tau) + T_{RPLgen} \quad (10)$$

$$T_{turnaround_{NBE_NACK}}(n, m, T_{RPLgen}, T_{TXprep}, T_{RXproc}, \tau) = 1.92 \text{ ms} + \frac{(n+m) \cdot 8}{250} \text{ ms} + 2 \cdot (T_{TXprep} + T_{RXproc} + \tau) + T_{RPLgen} \quad (11)$$

where, T_{CCA} denotes the time required for executing CSMA-CA, $T_{Datapkt}(x)$ signifies the time for sending the data frame with an x -byte payload, T_{ACK} indicates the time for sending the ACK frame and T_{RPLgen} denotes the time for processing the forward frame and generating the reply frame by the layers above the ones standardized by IEEE 802.15.4 (*e.g.*, the application layer).

For a nonbeacon-enabled PAN the minimum turnaround time for data transfer in CAP and CFP differs significantly. Besides, in CFP the turnaround time is affected by the duration of the GTSs and their direction (*i.e.*, whether the data within a GTS are transferred from the coordinator to a device or vice versa). During the CAP, the roundtrip time for acknowledged and unacknowledged scenarios is given by the following equations:

$$T_{turnaround_{CAP_ACK}}(n, m, T_{RPLgen}, T_{TXprep}, T_{RXproc}, \tau) = 3.296 \text{ ms} + \frac{(n+m) \cdot 8}{250} \text{ ms} + 2 \cdot (T_{TXprep} + T_{RXproc} + \tau) + T_{RPLgen} \quad (12)$$

$$T_{turnaround_{CAP_NACK}}(n, m, T_{RPLgen}, T_{TXprep}, T_{RXproc}, \tau) = 2.56 \text{ ms} + \frac{(n+m) \cdot 8}{250} \text{ ms} + 2 \cdot (T_{TXprep} + T_{RXproc} + \tau) + T_{RPLgen} \quad (13)$$

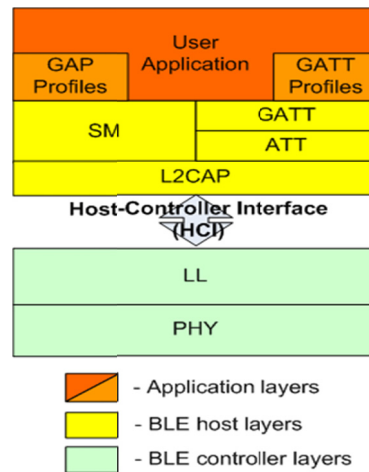
3.2. Bluetooth Low Energy

BLE was introduced as a part of Bluetooth Core Specification 4.0 in 2010 [3]. The major purpose of developing BLE was to enable products to have lower current consumption, lower complexity and lower cost than the ones possible with the classic Bluetooth [3].

Like the classic Bluetooth, a BLE transceiver includes two major components: a Controller and a Host [3]. The Controller is the logical entity that is responsible for the PHY layer and the link layer (LL) [3]. Although BLE Controllers inherit some features from the classic Bluetooth Controllers, both types of Controllers are not compatible [17]. The Host implements the functionalities of the upper layers. Those include (see Figure 4): L2CAP, GAP, ATT, GATT and SM. The logical link control and adaptation protocol (L2CAP) defines the procedures for higher level multiplexing, packet segmentation and transfer of quality of service (QoS) information [3]. The generic access profile (GAP) specifies generic procedures related to the discovery of devices, link establishment and termination management and procedures related to the use of the different security levels; it also includes the common format requirements of the parameters accessible on the user interface level. The security manager (SM) handles the management of pairing, authentication, bounding and encryption for BLE communication. The attribute protocol (ATT) specifies the mechanisms for discovering, reading and

writing attributes on a peer device, and the generic attribute profile (GATT) provides the framework for discovering services and for reading and writing characteristic values.

Figure 4. BLE stack architecture.



Like IEEE 802.15.4 with 2450 DSSS, BLE operates in the 2.4 GHz ISM band. Nonetheless, in order to reduce the transceivers' costs and the amount of energy consumed, BLE prescribes one to use binary frequency modulation with a 1 Mbit/s over-the-air data rate [3]. Unlike the classical Bluetooth, which uses 79 1-MHz-wide channels, BLE uses 40 2-MHz wide channels. Three of those channels, which are located between commonly used wireless local area network channels, are used for advertising and service discovery and are called *advertising channels*. The remaining 37 *data channels* are used to transfer the data. The transmission of data between BLE devices is bound to time units known as advertising and connection events [3].

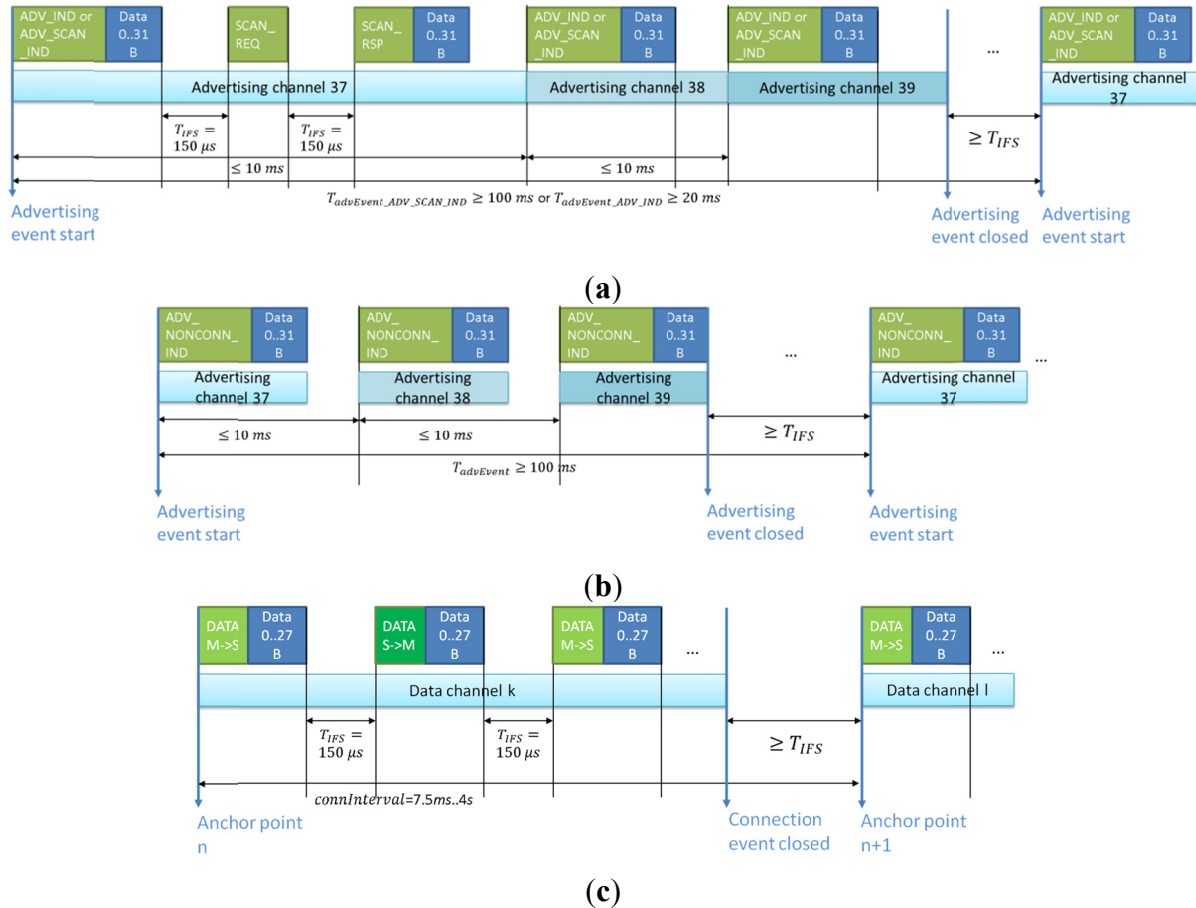
The advertising events are used to transmit data on the *advertising channels*. At the beginning of an advertising event an *advertiser*, i.e., the device that wants to transmit the data, sends an advertising frame. The format of the frame depends on the advertising event type. The standard [3] defines four possible types of advertising events (see Table 1) and the corresponding data unit format for the payload. Each advertising event consists of one or more advertising PDUs sent on the specific advertising channels. An advertising event is closed after one advertising PDU has been sent on each used advertising channel or earlier, if desired by the advertiser. In cases of ADV_IND or ADV_SCAN_IND advertising events, a device that listens to an advertising channel with no intention to connect to the advertiser (referred as *scanner*) can request more data from the advertiser by sending the SCAN_REQ PDU on the channel where the advertising channel PDU has been received (see Figure 5a). For ADV_IND or ADV_DIRECT_IND events, a device (referred to as an *initiator*) that listens to the channel and desires to exchange the data with the advertiser can send the CONNECT_REQ PDU. In that case, the advertiser and the initiator will try to establish a connection and start communication using *data channels*.

Table 1. Bluetooth Low Energy (BLE) advertising events.

Event type	Supported scanner responses	Minimum <i>advInterval</i> , ms	Maximum payload ¹ , bytes
ADV_IND	SCAN_REQ, CONNECT_REQ	20	31/31
ADV_DIRECT_IND	CONNECT_REQ		0
ADV_NONCONN_IND	-	100	31/-
ADV_SCAN_IND	SCAN_REQ	100	31/31

Notes: ¹ LL payload, format: N/M, N—maximum payload of an advertising frame, M—maximum payload of a SCAN_RSP.

Figure 5. (a) Timing for data transfer on BLE advertising channels using ADV_IND and ADV_SCAN_IND; (b) Timing for data transfer on BLE advertising channels using ADV_NONCONN_IND; (c) Timing for data transfer on BLE data channels.



For all undirected advertising events (*i.e.*, ADV_IND, ADV_NONCONN_IND or ADV_SCAN_IND) the time between two consecutive advertising events is defined as:

$$T_{advEvent} = advInterval + advDelay \quad (14)$$

The values of *advInterval* for different types of events are depicted in Table 1. The *advDelay* is a pseudo-random value ranging from 0 to 10 ms.

An advertising event ends and connection events start if the advertiser receives and accepts the connection request. Once a connection is established, the initiator becomes the master device and the advertiser becomes the slave device. As illustrated in Figure 5c, at the beginning of each connection event (referred to as the connection event *anchor point*) the used radio channel is changed following the predefined sequence. The communication in each connection event is initiated by the master device, which sends a frame to the slave. The master and the slave alternate sending the frames on the same data channel while at least one of the devices has data to transmit or until the end of the current connection event. In the case if either master or slave receives two consecutive frames with CRC errors, the connection event is closed. The same happens if either of the devices is missing a radio packet. According to the specification [3], the minimum time between frames on the same data channel should exceed the IFS, which equals 150 μ s. Note that unlike IEEE 802.15.4, which defines the IFS as the period of time between two successive frames transmitted from a device, BLE considers IFS the time interval between two consecutive frames on one channel.

The timing of connection events is determined using two parameters, namely the connection event interval (*connInterval*), and the slave latency (*connSlaveLatency*). The *connInterval* is a multiple of 1.25 ms and has values ranging from 7.5 ms to 4.0 s. The *connSlaveLatency* defines the maximum number of consecutive connection events in which a slave device is not required to listen to the master (used to enable energy saving). The format for BLE frames is depicted in Figure 3b.

3.2.1. Maximum Throughput on Advertising Channels

As illustrated in Figures 5a, 5b and Table 1, BLE enables data transmission in ADV_IND, ADV_SCAN_IND and ADV_NONCONN_IND advertising events. To include the data in the advertising frames, the Host sends the Controller either the LE_Set_Advertising_Data_Command or LE_Set_Scan_Response_Data_Command and specifies up to 31 data bytes to be included in the advertising or scan response frames. Note that upper layers of BLE add some overhead and each frame will carry less than 31 bytes of user-defined data. Nonetheless, to make the comparison with other technologies fair, we will consider the LL payload when discussing BLE. Assuming that in each of the advertising events the advertiser sends new data, the maximum throughput can be estimated as

$$Throughput_{BLE_{ADV}}(n) = \frac{n \cdot 8}{T_{advEvent}}, n = 0..31 \text{ bytes} \quad (15)$$

where $T_{advEvent}$ is defined from Equation (14) with *advInterval* value taken from Table 1.

Note that data transferring on the BLE advertising channels is not straightforward. The major challenge is that according to the standard [3], the BLE Controller has no means to inform the Host about the end of an advertising event so that the Host can update the advertising data. Therefore, to enable such data transmission, the application layer needs to control the timing on its own and periodically update the advertising data.

3.2.2. Maximum Throughput on Data Channels

The transfer of data using connection events and data channels is illustrated in Figure 5c. The throughput for master-slave unidirectional data transfer is given by

$$\text{Throughput}_{BLE_{DATA}}(n, m, T_{TXprep}, T_{RXproc}, \tau) = \frac{n \cdot 8}{0.16ms + \frac{(n+m) \cdot 8}{1000} + 2 \cdot \tau + 2 \cdot \max(0.15ms, T_{TXprep} + T_{RXproc})}, n, m = 0..27 \text{ bytes} \quad (16)$$

where n denotes the payload of each frame transferred from the master to the slave (*i.e.*, forward frame), m is the payload of the reply frame and 0.16 ms accounts for the time for transmitting the frame headers.

Note that in real-life scenarios, strong interferences and losses can significantly decrease the BLE throughput. The reason for this is that once a BLE transceiver misses a frame, communication is suspended until the next connection event. Moreover, if a device receives two consecutive frames with CRC errors, it will also suspend the transmission until the following event.

Clearly, if $n = 27, T_{TXprep} = T_{RXproc} = \tau = m = 0$ (16) gives a maximum BLE LL throughput of 319.5 kbit/s and considering that the overhead introduced by the upper layers of the forward frame protocols equals 7 bytes (see [17] and [18]), we obtain a maximum application layer throughput of 236.7 kbit/s. This corresponds to the analytic results reported in [17] and [18], which are obtained using Equations (1–3) in [18] for a no packet loss scenario.

3.2.3. Minimum Turnaround Time

For the transfer on data channels, the minimum turnaround time is

$$\begin{aligned} T_{turnaround_{BLE}}(n, m, T_{RPLgen}, T_{TXprep}, T_{RXproc}, \tau) \\ = 0.16ms + \frac{(n+m) \cdot 8}{1000} + 2 \cdot \tau + T_{TXprep} + T_{RXproc} + \max(0.15ms, T_{TXprep} \\ + T_{RPLgen} + T_{RXproc}) \end{aligned} \quad (17)$$

3.3. SimpliciTI

SimpliciTI is an open-source low-power proprietary radio protocol developed by TI for their wireless products (both the IEEE 802.15.4-compatible and the proprietary transceivers) [29]. The target of the protocol is to enable the fast and low-cost development of the low-power wireless networking applications using TI's products. Like the other proprietary protocols, SimpliciTI lacks rigid specification and is provided as a software (SW) stack with a set of examples and minimum documentation.

SimpliciTI is currently available for TI's CC1100/2500 (*i.e.*, 433/868/2,400 MHz radio transceivers supporting OOK, FSK, GFSK, MSK modulation), CC2420 (*i.e.*, 2,400 MHz radio transceiver compatible with IEEE 802.15.4) and the chips originating from those. The typical over-the-air data rate for SimpliciTI is 250 kbit/s, although users can easily change this as well as the other radio settings. Unlike IEEE 802.15.4, SimpliciTI enables users to choose whether they would like to use CCA or not. Moreover, SimpliciTI does not define IFS. The format of a SimpliciTI frame is depicted in Figure 3c.

3.3.1. Maximum Throughput

The maximum throughput for SimpliCI can be calculated by Equations (18) and (19) for the cases in which one uses CCA and when CCA is not required. In Equations (18), (19) and onwards $T_{RX \rightarrow TX}$ and $T_{TX \rightarrow RX}$ denote the time needed to switch between receive and transmit modes and vice versa, T_{CT} is the minimum time required to test the channel during CCA, DR stands for the used over-the-air data rate and N_{preamble} and N_{SW} are the lengths of preamble and synchroword in bytes, respectively. For SimpliCI the $T_{RX \rightarrow TX}$, $T_{TX \rightarrow RX}$ and T_{CT} are the hardware parameters of the transceiver, and DR, N_{preamble} and N_{SW} are set by users for the majority of transceivers.

$$\begin{aligned} \text{Throughput}_{\text{SITI}_{\text{CCA}}} (n, T_{\text{TXprep}}, T_{\text{RXproc}}, T_{\text{RX} \rightarrow \text{TX}}, T_{\text{TX} \rightarrow \text{RX}}, T_{\text{CT}}, \text{DR}, N_{\text{preamble}}, N_{\text{SW}}) \\ = \frac{n \cdot 8}{\frac{(n + 11 + N_{\text{preamble}} + N_{\text{SW}}) \cdot 8}{\text{DR}} + \max(T_{\text{TXprep}} + T_{\text{RX} \rightarrow \text{TX}} + T_{\text{TX} \rightarrow \text{RX}} + T_{\text{CT}}, T_{\text{RXproc}})} \end{aligned} \quad (18)$$

$$\text{Throughput}_{\text{SITI}_{\text{noCCA}}} (n, T_{\text{TXprep}}, T_{\text{RXproc}}, \text{DR}) = \frac{n \cdot 8}{\frac{(n + 11 + N_{\text{preamble}} + N_{\text{SW}}) \cdot 8}{\text{DR}} + \max(T_{\text{TXprep}}, T_{\text{RXproc}})} \quad (19)$$

3.3.2. Minimum Turnaround Time

The minimum turnaround time for SimpliCI can be calculated by:

$$\begin{aligned} T_{\text{turnaround}_{\text{SITI}_{\text{noCCA}}}} (n, m, T_{\text{RPLgen}}, T_{\text{TXprep}}, T_{\text{RXproc}}, \text{DR}, \tau) \\ = \frac{(n + m + 22 + 2 \cdot (N_{\text{preamble}} + N_{\text{SW}})) \cdot 8}{\text{DR}} + 2 \cdot (T_{\text{TXprep}} + T_{\text{RXproc}} + \tau) + T_{\text{RPLgen}} \end{aligned} \quad (20)$$

$$\begin{aligned} T_{\text{turnaround}_{\text{SITI}_{\text{CCA}}}} (n, m, T_{\text{RPLgen}}, T_{\text{TXprep}}, T_{\text{RXproc}}, T_{\text{RX} \rightarrow \text{TX}}, T_{\text{TX} \rightarrow \text{RX}}, T_{\text{CT}}, \text{DR}, \tau) \\ = T_{\text{turnaround}_{\text{SITI}_{\text{noCCA}}}} (n, m, T_{\text{RPLgen}}, T_{\text{TXprep}}, T_{\text{RXproc}}, \text{DR}, \tau) + 2 \cdot (T_{\text{RX} \rightarrow \text{TX}} + T_{\text{TX} \rightarrow \text{RX}} + T_{\text{CT}}) \end{aligned} \quad (21)$$

4. Experiment Methodology

In order to compare the analytic estimations of the maximum throughput and the minimum frame turnaround time obtained in Section 3 with the performance characteristics of the real-life transmitters, we have executed a set of experiments. In those, we have used the CC2430, CC2510 and CC2540 commercial Systems-on-Chips (SoCs) from TI (see Figure 6). The features of the transceivers are summarized in Table 2. For our tests, we developed a special measurement application layer that operated on top of MAC layers implemented by SimpliCI (version 1.2) and TIMAC (stack version 1.0.1 implementing IEEE 802.15.4) software stacks and on top of the Host Controller Interface (HCI) provided by TI-BLE software stack (version 1.2.1).

In our experiments we did the following:

- defined the maximum unidirectional LL data throughput for the HW and SW implementations of the protocols;
- measured the throughput and the energy consumption of the transceivers;
- measured the minimum turnaround time;
- measured the resources required to implement the protocols.

Figure 6. Hardware modules used for the tests: the front row includes extension radio boards (from left to right) CC2540 (BLE), CC2430 (IEEE 802.15.4) and CC2510 (SimpliciTI); the back row includes the battery extender board (left) and the SmartRF04 development board (right).

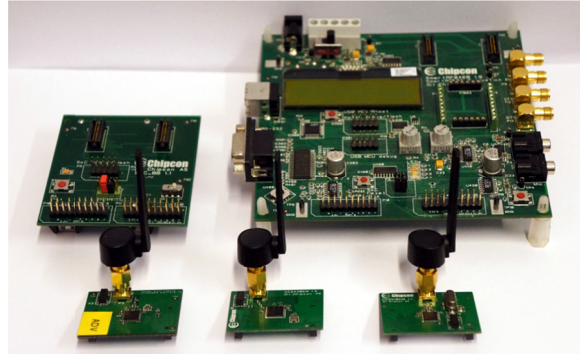


Table 2. Nominal parameters and settings of the radio transceivers used for the tests.

Parameter	Device		
	CC2510	CC2431	CC2540
Device type	system-on-chip (radio + 8051 microcontroller)	system-on-chip (radio + 8051 microcontroller)	system-on-chip (radio + 8051 microcontroller)
Microcontroller specification	Clock: 26 MHz	Clock: 32 MHz	Clock: 32 MHz
	Flash: 32 kbyte	Flash: 128 kbyte	Flash: 256 kbyte
	RAM: 4 kbyte	RAM: 8 kbyte	RAM: 8 kbyte
Radio protocol and stack version	SimpliciTI (TI SimpliciTI v1.2)	IEEE 802.15.4 (TI-MAC v1.0.1)	BLE (TI-BLE v1.2.1)
Frequency band	2.4 GHz	2.4 GHz	2.4 GHz
Modulation	MSK	O-QPSK	GFSK
Spectrum spreading	None	DSSS	FHSS
Over-the-air data rate, kbit/s	250/500	250	1000
TX power range, dBm	−55...1	−25.2...0.6	−23...4
RX sensitivity, dBm	−90 (at 250 kbit/s over-the-air data rate)	−92	−87 (at standard mode)
Supply voltage, V	2–3.6	2–3.6	2–3.6
Sleep current consumption, μA	0.5	0.5	0.9 ¹
Price (normalized)	0.358	1	0.341

Note: ¹ timer active.

The measurements of the throughput and turnaround time were conducted in the laboratory environment using the channel with minimum interference for SimpliciTI and IEEE 802.15.4 (the absence of interference from other systems was confirmed prior to the measurements). The measurements were executed with the transmitter placed at a distance of one meter away from the receiver while using the maximum possible radio transmitting power. This ensured that the strength of the received radio signal was well above the transceiver's sensitivity level (typically, the received signal strength indicator (RSSI) was well above −40 dBm and the packet error rate was below 0.5%).

For the throughput estimation, in order to decrease possible environmental effects we measured the total time required for the transmission of 60,000 data frames with a predefined payload size from the transmitter to the receiver. The turnaround time was estimated by averaging the measurements of the turnaround time for 1,000 data frames sent to and from the transmitter.

To measure the power consumption, we used the current-shunt method (refer to [30–32]). The maximum error for these measurements is below 3 mW. The power consumption of the transceivers supplied from the laboratory power source was measured. The supply voltage was set to 3 V. While measuring the power consumption, all the peripherals on the used test boards were powered down—the only active components were the radio transceiver and the microcontroller core running the stacks.

As shown in Table 2, all the used transceivers have a processing core based on the 8051 microcontroller. Therefore, by measuring the resources required to implement the protocols we were able to get a sufficiently fair estimation of the protocols' complexity. Note that the developed application layers had the same functionality and were compiled using the same compiler with identical optimization settings.

5. Discussion

The data presented in Table 3 reveal the amount of program and data memory required to implement the stacks. As shown, the complete BLE stack requires almost four times more program memory than TIMAC and more than eight times more resources than SimpliCI. The implementation of the PHY, LL and HCI BLE layers required 1.5 times more resources than the whole TIMAC stack and 3.5 times more resources than SimpliCI.

Table 3. Resources consumed by the stacks.

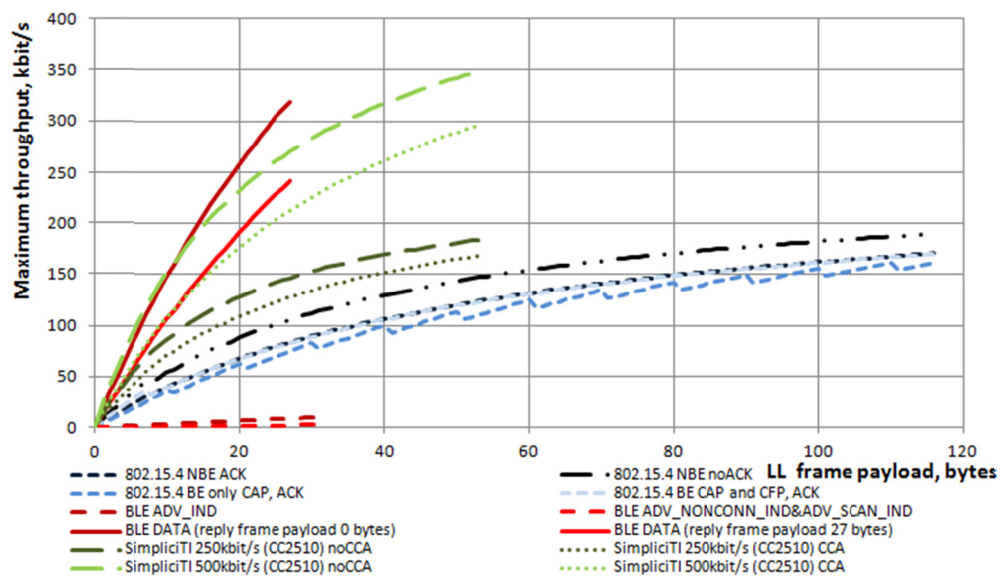
Resource	Stack			
	SimpliCI	TI-MAC	TI-BLE (Master)	TI-BLE (Slave)
Program memory, bytes	16,024	36,573	55,786/137,719 ²	50,913/117,050 ²
Data memory ¹, bytes	3,567	5,438	10,400/12,750 ²	9,082/10,676 ²

Note: ¹ Cumulative for RAM and Flash; ² For PHY, LL and test application layers on top of HCI only/ for the complete stack including PHY, LL, HCI, L2CAP, GAP, ATT, GATT, GATT and GAP profiles.

The analytic estimations of the maximum possible throughput of the three technologies for the different frame payloads are depicted in Figure 7. The results have been obtained using Equations (3–9), (15), (16), (18), (19) assuming that $T_{TXprep} = T_{RXproc} = \tau = 0$, $T_{RX \rightarrow TX} = 32.6 \mu s$, $T_{TX \rightarrow RX} = 33.6 \mu s$, $T_{CT} = 1000 \mu s$, $N_{SW} = 4$ bytes, and $N_{preamble} = 4$ bytes for $DR = 250$ kbit/s and $N_{preamble} = 8$ bytes for $DR = 500$ kbit/s (see [33]). The presented results reveal that although BLE has the highest over-the-air data rate among all protocols, SimpliCI using the $DR = 500$ kbit/s can potentially provide a higher throughput. The major reasons for this are the following. First of all, during unidirectional data transmission, SimpliCI does not have to send any data from the receiver node to the transmitter, whilst the BLE receiver always has to send a frame in reply to each received frame to continue communication in an event (see Section 3.2). Second, unlike the two other protocols, SimpliCI does not define the IFS between transmitted frames. Third, the payload in SimpliCI

frames is about two times larger than the one possible in a BLE data frame. The presented results also reveal that the maximum throughput possible for this protocol is 1.5 to 2 times lower than that of BLE and SimpliCI TI, even though the IEEE 802.15.4 frames are capable of carrying the highest payload. The major reasons for this are as follows: the lower over-the-air data rate compared with the other technologies (see Table 2), the mandatory use of CCA before each transmission in the nonbeacon-enabled mode and in CAP of the beacon-enabled mode, and the protracted IFS between the subsequent frames. In addition, Figure 7 reveals that the maximum possible throughput for data transfer using BLE advertising channels is below 10 kbit/s for ADV_IND events and is below 2.4 kbit/s for ADV_NONCONN_IND and ADV_SCAN_IND events. Note that the presented value for BLE's maximum throughput is calculated for the LL payload and the actual throughput for the user applications data is lower.

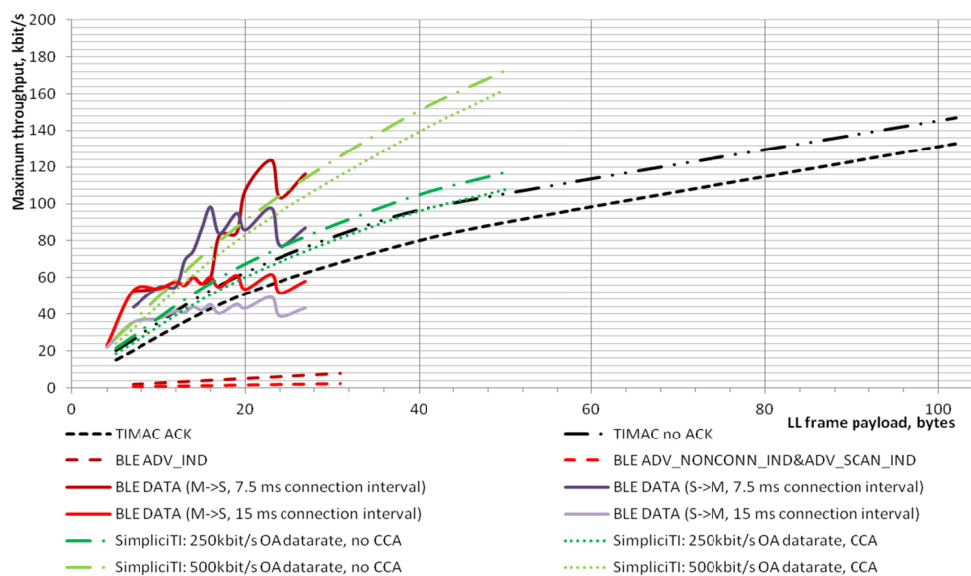
Figure 7. LL frame payload's effect on the maximum unidirectional single-hop throughput for IEEE 802.15.4, BLE and SimpliCI TI (analytic results).



The maximum throughput obtained in the experiment using real transceivers is depicted in Figure 8. Comparing Figure 8 with Figure 7 one can note that the maximum real-life throughput obtained using SimpliCI TI is about two times lower than the theoretically expected one. The major reason for this is the time required for generating a frame, writing it to the transceiver's memory and executing all required service operations before the transmission (e.g., frequency synthesizer calibration). The maximum throughput, obtained with TIMAC stack and CC2430 transceivers, reaches 145 kbit/s for unacknowledged and 134 kbit/s for acknowledged data transmission. This is approximately 20%–25% lower than the maximum throughput possible for IEEE 802.15.4 according to the analysis (see Figure 7) and the practical experiments reported in [6]. The reasons for this are the maximum MAC payload's restriction in TIMAC to 102 bytes (refer to [34]), the delays in generating and copying the frames to the transceiver's buffer, and the specifics of the CCA and IFS implementation within TIMAC (e.g., the TIMAC does not support the execution of CCA during the IFS). In [20] the authors noted that the TI BLE stack restricts the number of data frames sent in a connection event to four. In our experiments, we have seen that although the stack limits the amount of data transferred in a single

connection event, this is done based on the amount of data rather than the number of frames. Therefore, for frames with 24–27 byte payloads, only four frames are transmitted from the master to the slave (and only three frames vice versa) in a connection event, regardless of the connection interval. For the frames with 20–23 byte payloads, five frames are sent from the master to the slave and four from the slave to the master in a single connection event. As we decreased the frame payload, the number of the frames sent in a connection event increased. Nonetheless, we noticed that once we decreased the payload below some level (which depends on the used connection interval), the data were exchanged only in one out of each set of two consecutive connection events. We expect that this was caused by the erroneous estimation of the data transfer duration by the stack's LL, which caused the master node to miss the start of the next event. These are the two major reasons the measured maximum BLE LL throughput was 122.6 kbit/s, which is 2.6 times lower than the theoretical maximum. Nonetheless, comparing the results of our BLE maximum throughput measurements to the results reported in previous works (e.g., the reported in [17] 58.5 kbit/s for the application layer's throughput, which corresponds to the throughput of 79 kbit/s at the LL) one can see that in our experiment we obtained slightly higher values for the maximum throughput. The two major reasons for this are the use of lower payload values, which enabled us to increase the number of packets sent in one connection event, and the lower packet processing delays due to exclusion of the BLE stack layers above the HCI. For the data transfer on BLE advertising channels the maximum measured throughput was around 8 kbit/s (advertising data are changed every 31 ms) for ADV_IND and 2.2 kbit/s (data are changed after 111 ms) for ADV_NONCONN_IND and ADV_SCAN_IND events, respectively.

Figure 8. LL frame payload's effect on the maximum unidirectional single-hop throughput for IEEE 802.15.4, BLE and Simpliciti (experimental results).



The analytic estimations of the minimum turnaround time that were calculated from Equations (10), (11), (17), (20) and (21) for $m = 0$ and $T_{RPLgen} = 0$ are presented in Figure 9. As one can see, BLE is expected to have the lowest turnaround time among the protocols under discussion. This is not surprising, as BLE enables the receiver to send a reply frame immediately after the IFS, which equals $150 \mu s$. Simpliciti is expected to have a roundtrip time of 0.7 ms to 2.9 ms for the transmissions at

DR = 500 kbit/s and 1.2 ms to 5 ms at $DR = 250$ kbit/s. The roundtrip time for IEEE 802.15.4 is estimated to range from 1.92 ms to 9.34 ms for unacknowledged and from 2.65 ms to 10.08 ms for acknowledged data transfer, depending on the payload size.

Figure 9. LL frame payload's effect on the minimum single-hop turnaround time for IEEE 802.15.4, BLE and SimpliCI TI (analytic results).

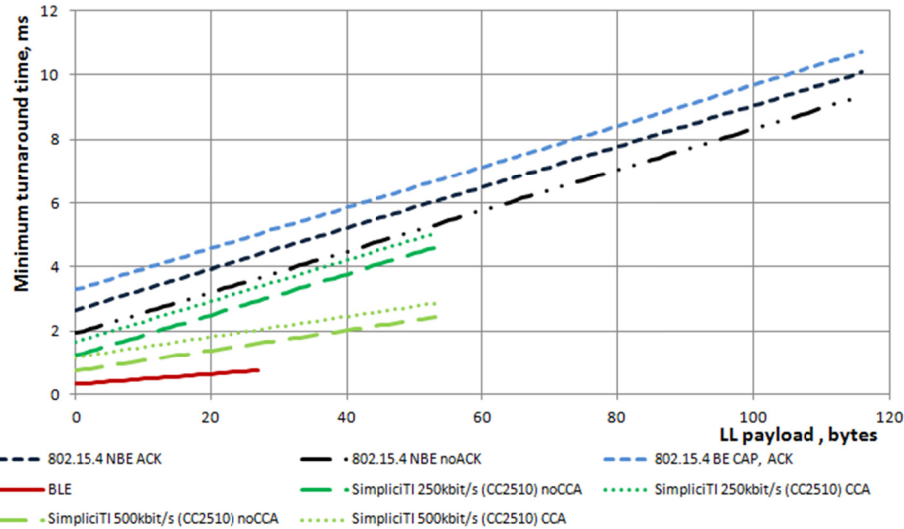


Figure 10. LL frame payload's effect on the minimum single-hop turnaround time for IEEE 802.15.4, BLE and SimpliCI TI (experimental results).

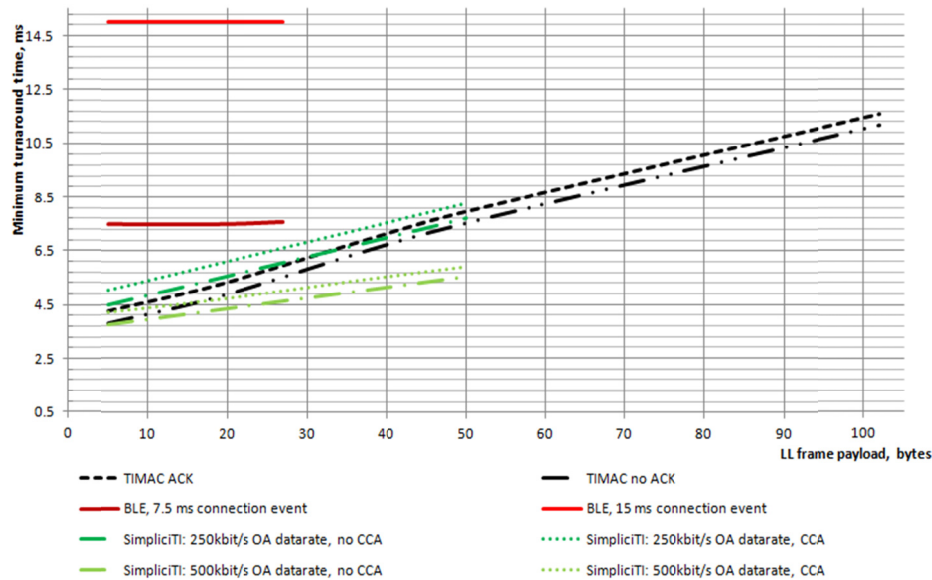
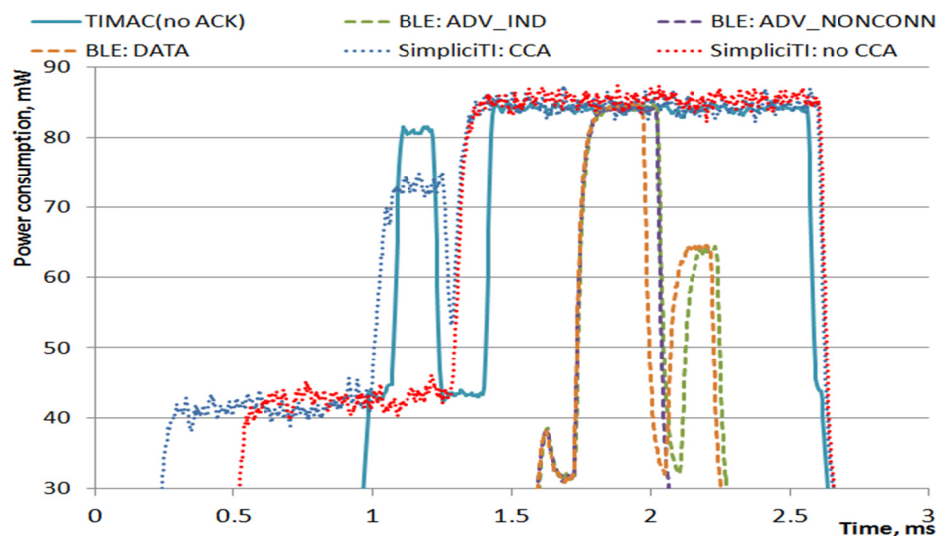


Figure 10 depicts the measured turnaround time for real-life transceivers. The presented data reveal that the real-life turnaround time for IEEE 802.15.4 and SimpliCI TI transceivers is around 1.5–3 ms higher than the analytic expectations. Again, we believe that this discrepancy is caused by the time required for generating the data frames and executing all the required service operations. Meanwhile, for BLE, instead of the turnaround time being less than one millisecond, in the tested system we

observed the turnaround time to be slightly above the connection interval duration. The reason for this is that the TI BLE stack delays transmission of the reply until the start of the next connection event.

Figure 11. Power consumption for the tested transceivers (see Table 2) and protocols for transmitting a 19-byte frame (supply voltage 3 V, radio transmit power 0 dBm).



The results revealing the tested transceivers' energy consumption are presented in Table 4 and depicted in Figure 11. The results reveal that the BLE transceivers require much less time and energy for sending the data than the IEEE 802.15.4 and SimplicTI transceivers. This happens even though BLE transmission includes both the transmission and reception phases (see Figures 5c and 11). The results in Table 4 for the energy consumption for different protocols for transferring frames with 19-byte payloads reveal that the BLE radios require two to seven times less energy compared with the other tested protocols. This result corresponds quite well to the results presented in [20]. When the energy consumption of BLE is compared with the other protocols using higher frame payloads (see Table 4), we observe that even in this case the amount of energy required to transfer the data over BLE is two to three times lower.

Table 4. Required time and consumed energy for frame transmission/reception for the transceivers.

Stack	Time ¹ , ms	Consumed energy ¹ , μJ	Energy efficiency ¹ , μJ/byte
<i>Transmission (for BLE - also reception) of a single frame with a 19-byte LL payload:</i>			
TIMAC, acknowledged ²	2.50	190	10.0
TIMAC, unacknowledged	1.66	125	6.6
BLE, ADV_IND ³	0.73	42	2.2
BLE, ADV_NONCONN_IND	0.50	31	1.6
BLE (master node), data frame transmission ⁴	0.66	39	2.1
BLE (master node), data frame reception ⁵	0.66	36	1.9
SimpliciTI, CCA, 250 kbit/s	2.46	165	8.7
SimpliciTI, no CCA, 250 kbit/s		2.21	148
SimpliciTI, CCA, 500 kbit/s		1.76	105
SimpliciTI, no CCA, 500 kbit/s		1.60	96

Table 4. Cont.

Stack	Time ¹ , ms	Consumed energy ¹ , μJ	Energy efficiency ¹ , μJ/byte
<i>Transmission (for BLE - also reception) of a single frame with other payloads:</i>			
TIMAC, acknowledged ² , 100-byte payload		5.07	406 4.1
TIMAC, unacknowledged, 100-byte payload		4.30	347 3.5
BLE, ADV_IND ³ , 31-byte payload		0.80	50 1.6
BLE, ADV_NONCONN_IND, 31-byte payload		0.60	39 1.3
BLE (master node), 27-byte data frame transmission ⁴		0.72	44 1.6
BLE (master node), 27-byte data frame reception ⁵		0.72	40 1.5
SimpliciTI, 250 kbit/s, CCA, 50-byte payload		3.5	246 4.9
SimpliciTI, 250 kbit/s, no CCA, 50-byte payload		3.16	227 4.5
SimpliciTI, 500 kbit/s, CCA, 50-byte payload		2.23	148 3.0
SimpliciTI, 500 kbit/s, no CCA, 50-byte payload		2.09	141 2.8

Note: ¹ Energy and time for pre-processing and post-processing (see [19]) not included; ² Reception of an acknowledgement included; ³ Energy and time for checking the channel for incoming SCAN_REQ included; ⁴ Energy and time for receiving the reply frame with no payload included; ⁵ Energy and time for sending the starting frame at the beginning of a connection event with no payload included (see Figure 5c).

6. Conclusions

Bluetooth Low Energy (BLE) is rather new protocol, and in this study we deepened the understanding of both the theoretical capabilities of the protocol itself and the capabilities of the currently available transceivers to implement the protocol. In the paper, we also compared BLE with two other protocols-SimpliciTI, which is a proprietary protocol developed by Texas Instruments, and IEEE 802.15.4, which is the de-facto communication standard in the WSNs.

The results reveal that BLE can potentially support the maximum LL data throughput of around 320 kbit/s. This is about 70% higher than the maximum throughput possible for IEEE 802.15.4 with 2450 DSSS PHY (*i.e.*, 190 kbit/s). Nonetheless, the results of the measurements that were executed using the real BLE transceivers revealed that the current version of the used BLE stack has several limitations, which prevented us from obtaining a throughput higher than 123 kbit/s. This is about 20% lower than the throughput we have obtained with the IEEE 802.15.4 transceivers and 40% lower than the maximum throughput that was measured for SimpliciTI transceivers. The maximum throughput that we managed to obtain on BLE advertising channels was below 10 kbit/s. The absence of the mechanism for the BLE Controller to inform the Host about the start/end of an advertising event complicates the implementation of the data transfer on BLE advertising channels.

The presented analytic results show that the BLE technology is capable of providing a frame turnaround time of less than one millisecond. The analytic estimations of the minimum turnaround times for SimpliciTI and IEEE 802.15.4 range from 0.7 ms to 5 ms and from 2.7 ms to 10 ms, respectively. The turnaround time that we have measured using SimpliciTI and the IEEE 802.15.4 transceivers appeared to be 1.5–3 ms higher than the analytic expectations. The major reason for this discrepancy is the time required to generate the data frames and to execute the other service operations. This time depends exclusively on the features of the hardware and software that implement the protocol and was not accounted for during the analysis. The tested BLE transceivers delayed the

transmission of the reply data frame until the start of the next connection event. Due to this feature of the BLE stack, the minimum frame turnaround time that we observed during our experiments was around 7.6 ms.

The conducted experiments have shown that the tested BLE transceiver required two to seven times less energy to transfer data than the SimpliciTI and the IEEE 802.15.4 transceivers. This result corresponds to the results presented in [20].

Table 2 reveals that the price of the BLE transceiver chip is slightly lower than the price of the SimpliciTI chip and about three times lower than the cost of the IEEE 802.15.4 transceiver. Additionally, the presented results have shown that the BLE software stack requires significantly more resources than SimpliciTI or IEEE 802.15.4. The implementation of the complete BLE stack requires almost four times more program memory than was used by TIMAC (implementing IEEE 802.15.4) and more than eight times more than was necessary for SimpliciTI.

One of the factors that can somewhat limit the applicability of BLE is the restrictions concerning the BLE network's topology. The current version of the standard requires a single-hop star network topology and states that a BLE device 'is only permitted to belong to one piconet at a time' [3]. Although this requirement does not forbid the establishment of the multihop links directly, it makes the implementation of multihopping for BLE more challenging. Indeed, to implement a multihop data transmission, a node will have to periodically switch the piconets to relay the data between those.

The other important issue is the interoperability between the BLE and the other telecommunication systems, many of which are based on the Internet Protocol (IP). To address this issue, the authors of [35] specified a mechanism enabling IPv6 transmission over BLE links. In [36] the details of the implementation and evaluation of the IPv6 packets transmission over BLE are reported. Although the mechanism suggested in [35] enables BLE devices to transfer the IPv6 packets, it also requires the devices to support packet fragmentation at L2CAP (which is optional for BLE [3]) and to have a buffer capable of storing at least a datagram of at least 1280 bytes [37]. It is also not clear how energy efficient it is to use the IPv6 over BLE links and which applications require this capability, especially considering the restrictions regarding the topology of the BLE networks.

Nonetheless, BLE has two significant advantages: its interoperability with other devices and its low cost. One of the factors that impeded the spreading of WSN-based applications is their absence of common communication interfaces enabling other devices (e.g., laptops, smartphones, palmtop computers and mobile devices) to communicate directly with WSNs. The Bluetooth Special Interest Group (SIG) is addressing this challenge by suggesting dual-mode transceivers, *i.e.*, radio modules that can support both the classic Bluetooth and BLE. The first consumer devices (*i.e.*, smart phones, tablet computers and notebooks) encapsulating such transceivers have already appeared on the market. Moreover, in this paper we have shown that the BLE transceivers are lower in price than transceivers implementing the other evaluated technologies. This enables one to reduce the cost of the applications developed using this technology.

To sum up, the results presented in this paper reveal that BLE already provides an inexpensive and power-efficient solution for wireless communication. Nonetheless, the tested BLE radio transceiver and stack still have many limitations that restrict the throughput and communication latency one can achieve with them. The other serious limitation of the BLE technology is the restrictions regarding its network topology.

Based on the analytic and measurement evaluation results, we expect that the major application area for BLE in the coming years will be energy efficient human-oriented applications that require either peer-to-peer or single-hop star topologies. These applications may include health and fitness, entertainment, smart home/office, personal security and proximity detection, and data and location advertisement. Although there is the mechanism enabling the IPv6 to be transferred over BLE links, we hardly think that this capability will be widely used in real-life applications in the immediate future. Meanwhile, we suppose that for non-human oriented applications and for applications requiring wide coverage areas (e.g., the wildlife, nature and environment monitoring, industrial monitoring and control, building and process automation, security, logistics) other communication protocols and standards will be predominantly used, such as IEEE 802.15.4 and its numerous extensions, e.g. ZigBee [21], WirelessHART [38], ISA100.11a [39] or 6LoWPAN [40]. Therefore, we suppose that BLE will have a wide application area, specifically for WSN applications, although this technology will hardly dominate the market by 2015 as the authors in [4] assumed.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Silicon Laboratories Inc. The evolution of wireless sensor networks Available online: <http://www.silabs.com/Support%20Documents/TechnicalDocs/evolution-of-wireless-sensor-networks.pdf> (accessed on 24 June 2013).
2. Hatler, M. Industrial wireless sensor networks: Trends and developments. Available online: <http://www.isa.org/InTechTemplate.cfm?template=/ContentManagement/ContentDisplay.cfm&ContentID=90824> (accessed on 24 June 2013).
3. Bluetooth SIG. *Bluetooth Specification Version 4*; The Bluetooth Special Interest Group: Kirkland, WA, USA, 2010.
4. *WTRS Wireless Sensor Network Technology Trends Report*; WT062510CNTS; West Technologies Research Solutions: Mountain View, CA, USA, 2010; pp. 1–267.
5. Sun, T.; Chen, L.J.; Han, C.C.; Yang, G.; Gerla, M. Measuring Effective Capacity of IEEE 802.15.4 Beaconless Mode. In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'06), Las Vegas, NV, USA, 3–6 April 2006; pp. 493–498.
6. Mikhaylov, K.; Tervonen, J. Analysis and evaluation of the maximum throughput for data streaming over IEEE 802.15.4 wireless networks. *J. High Speed Netw.* in press.
7. Latré, B.; Mil, P.D.; Moerman, I.; Dhoedt, B.; Demeester, P.; Dierdonck, N.V. Throughput and Delay Analysis of Unslotted IEEE 802.15.4. *J. Netw.* **2006**, *1*, 20–28.
8. Choi, J.S.; Zhou, M. Performance Analysis of ZigBee-Based Body Sensor Networks. In Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC'10), Istanbul, Turkey, 10–13 October 2010; pp. 2427–2433.
9. Liang, X.; Balasingham, I. Performance Analysis of the IEEE 802.15.4 Based ECG Monitoring Network. In Proceedings of IASTED Wireless and Optical Communications Conference (WOC'07); Montreal, QC, Canada, 30 May–1 June 2007; pp. 99–104.

10. Zhang, Y.; Atac, A.; Liao, L.; Heinen, S. A Low-Power High-Efficiency Demodulator in Bluetooth Low Energy Receiver. In Proceedings of 8th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME'12), Aachen, Germany, 12–15 June 2012; pp. 1–4.
11. Masuch, J.; Delgado-Restituto, M. A 190-microWatt Zero-IF GFSK Demodulator with a 4-b Phase-Domain ADC. *IEEE J. Solid-St. Circ.* **2012**, *47*, 2796–2806.
12. Wong, A.; Dawkins, M.; Devita, G.; Kasparidis, N.; Katsiamis, A.; King, O.; Lauria, F.; Schiff, J.; Burdett, A. A 1 V 5 m A Multimode IEEE 802.15.6 / Bluetooth Low-Energy WBAN Transceiver for Biotelemetry Applications. In Proceedings of IEEE International Solid-State Circuits Conference (ISSCC'12), San Francisco, CA, USA, 19–23 February 2012; pp. 300–302.
13. Yu, B.; Xu, L.; Li, Y. Bluetooth Low Energy (BLE) Based Mobile Electrocardiogram Monitoring System. In Proceedings of International Conference on Information and Automation (ICIA'12), Shenyang, China, 6–8 June 2012; pp. 763–767.
14. Ali, M.; Albasha, L.; Al-Nashash, H. A Bluetooth Low Energy Implantable Glucose Monitoring System. In Proceedings of 8th European Radar Conference (EuRAD'11), Manchester, UK, 12–14 October 2011; pp. 377–380.
15. Jara, A.J.; Fernandez, D.; Lopez, P.; Zamora, M.A.; Ubeda, B.; Skarmeta, A.G. Evaluation of Bluetooth Low Energy Capabilities for Continuous Data Transmission from a Wearable Electrocardiogram. In Proceedings of 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'12), Palermo, Italy, 4–6 July 2012; pp. 912–917.
16. Liu, J.; Chen, C.; Ma, Y. Modeling Neighbor Discovery in Bluetooth Low Energy Networks. *IEEE Commun. Lett.* **2012**, *16*, 1439–1441.
17. Gomez, C.; Oller, J.; Paradells, J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology. *Sensors* **2012**, *12*, 11734–11753.
18. Gomez, C.; Demirkol, I.; Paradells, J. Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link. *IEEE Commun. Lett.* **2011**, *15*, 1187–1189.
19. Kamath, S.; Lindh, J. *Measuring Bluetooth® Low Energy Power Consumption*; AN092S (WRA347a); Texas Instruments, Inc.: Dallas, TX, USA, 2012; pp. 1–24.
20. 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 IEEE Wireless Communications and Networking Conference Workshops (WCNCW'12), Paris, France, 1 April 2012; pp. 232–237.
21. *ZigBee Specification*; 053474r17; ZigBee Standards Organization: San Ramon, CA, USA, 2008; pp. 1–576.
22. Fujii, C.; Seah, W.K.-G. Multi-Tier Probabilistic polling in Wireless Sensor Networks Powered by Energy Harvesting. In Proceedings of 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'11), Adelaide, SA, USA, 6–9 December 2011; pp. 383–388.
23. Skrzypczak, L.; Grimaldi, D.; Rak, R. Basic Characteristics of ZigBee and Simplici Modules to Use in Measurement Systems. In Proceedings of 19th IMEKO World Congress, Lisbon, Portugal, 6–11 September 2009; pp. 1456–1460.

24. Bertarelli, F. Energy Cluster Aggregation in a WSN Based on EZ430-RF2500 T Nodes and SimpliciTI Protocol. In Proceedings of 4th Education and Research Conference (EDERC'10), Nice, France, 1–2 December 2010; pp. 145–149.
25. IEEE 802 Working Group. *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*; IEEE Std 802.15.4-2003; 2003; Volume 4, pp. 1–670.
26. IEEE 802 Working Group. *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)*; IEEE Std 802.15.4-2006; 2006; pp. 1–320.
27. IEEE 802 Working Group. *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*; IEEE Std 802.15.4-2011; 2011; pp. 1–314.
28. Aaberge, T. Low Complexity Antenna Diversity for IEEE 802.15.4 2.4 GHz PHY. M.Sc. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, June 2009.
29. Texas Instruments SimpliciTI™ - RF Made Easy. Available online: <http://www.ti.com/simpliciti/> (accessed on 14 June 2013).
30. Nakutis, Z. Embedded Systems Power Consumption Measurement Methods Overview. *MATAVIMAI* **2009**, *2*, 29–35.
31. Mikhaylov, K.; Tervonen, J. Optimization of Microcontroller Hardware Parameters for Wireless Sensor Network Node Power Consumption and Lifetime Improvement. In Proceedings of 2nd International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT'10), Moscow, Russia, 18–20 October 2010; pp. 1150–1156.
32. Mikhaylov, K.; Tervonen, J. Evaluation of Power Efficiency for Digital Serial Interfaces of Microcontrollers. In Proceedings of 5th International Conference on New Technologies, Mobility and Security (NTMS), Istanbul, Turkey, 7–10 May 2012; pp. 1–5.
33. Texas Instruments. *Low-Power SoC (System-on-Chip) with MCU, Memory, 2.4 GHz RF Transceiver, and USB Controller*; 2013; p. 236.
34. Texas Instruments. *802.15.4 MAC Application Programming Interface*; SWRA192; 2009; p. 58.
35. Isomaki, M.; Nieminen, J.; Gomez, C.; Shelby, Z.; Savolainen, T.; Patil, B. Transmission of IPv6 Packets over BLUETOOTH Low Energy. Available online: <http://tools.ietf.org/html/draft-ietf-6lowpan-btle-12#section-2.4> (accessed on 1 August 2013).
36. Wang, H.; Xi, M.; Liu, J.; Chen, C. Transmitting IPv6 Packets over Bluetooth Low Energy Based on BlueZ. In Proceedings of 15th International Conference on Advanced Communication Technology (ICACT'13), PyeongChang, Korea, 27–30 January 2013; pp. 72–77.
37. Deering, S.E.; Hinden, R. Internet Protocol, Version 6 (IPv6) Specification. Available online: <http://tools.ietf.org/html/rfc2460> (accessed on 1 August 2013).
38. *WirelessHART*; IEC 62591:2010(E); International Electrotechnical Commission: Austin, TX, USA, 2010.
39. *Wireless Systems for Industrial Automation: Process Control and Related Applications*; ANSI/ISA-100.11a-2011; International Society of Automation: Research Triangle Park, NC, USA, 2011; pp. 1–792.

40. Kushalnagar, N.; Montenegro, G.; Culler, D.E.; Hui, J.W. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Available online: <http://tools.ietf.org/html/rfc4944> (accessed on 2 August 2013).

© 2013 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 license (<http://creativecommons.org/licenses/by/3.0/>).