# Simulation of Network-Level Performance for Bluetooth Low Energy

Konstantin Mikhaylov
Centre for Wireless Communications
University of Oulu, P.O. BOX 4500, Oulu, Finland
Email: konstantin.mikhaylov@ee.oulu.fi

***Abstract*—Bluetooth Low Energy (BLE) is a recently developed energy-efficient short-range radio communication technology, which is currently vastly gaining popularity and which might become an enabler for implementing in practice the Internet of Things (IoT) concept. However, the complexity of the protocol and the lack of the simulation tools impedes the studies of BLE network performance. For this reason, in the paper we introduce the novel simulation tool, which is capable of simulating the communication in the networks of BLE devices. In the paper we discuss the capabilities and limitations of the suggested tool and compare the results of the simulations with the ones obtained with the real-life hardware BLE transceivers. Also, the paper highlights some of the results obtained with the developed tool, which disclose the effect of the communication parameters on the performance of the BLE protocol in the multi-node scenarios and point out few weak points of the BLE technology.**

***Index Terms*—Bluetooth Low Energy, Bluetooth Smart, BLE, network, performance, simulation.**

## I. Introduction

Bluetooth Low Energy (BLE) is one of the recently developed energy-efficient short-range wireless communication protocols. The protocol has been standardized in 2010 as the part of Bluetooth Core Specification version 4.0 [1]. The major purpose of developing the protocol was to enable transceivers with lower power consumption, lower complexity and lower price than the ones possible with the classic Bluetooth [1], [2]. Although BLE operates in the same 2.4 GHz Industrial, Scientific and Medical (ISM) band and inherits few mechanisms from the classical Bluetooth, many new technical solutions were introduced in order to reduce the power consumption. Those made the two technologies not compatible [3], [4]. One of the major advantages of BLE over the other protocols is the compatibility with a large number of commercial mobile phones, tablets and computers (called ”Bluetooth Smart Ready” [5]).The availability of BLE in those provides a good starting point for developing the various Bluetooth Smart sensors and actuators to be coupled with Bluetooth Smart Ready devices.

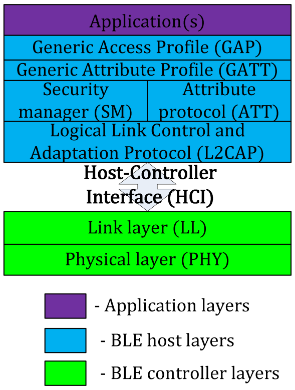

Fig. 1. BLE stack

Even though the first version of the BLE standard has been introduced in mid 2010, the technology got serious attention from the academy quite recently. Multiple medicine-oriented applications using BLE as the communication technology were reported by the authors in [6]–[10]. The performance of the protocol and of the currently available hardware BLE transceivers was discussed in [3], [4], [11]–[15]. In [3], [11] it was shown that the maximum theoretical throughput of a peer-to-peer (P2P) BLE link at the application layer is 236.7 kbit/s (or 319.5 kbit/s at Link Layer (LL) according to [4]). In [3], [4] it was shown that the throughput of the current commercial BLE transceivers is well below the theoretic maximum. In [4], [12]–[15] it was reported that in terms of energy efficiency, BLE outperforms the state-of-art technologies (e.g., IEEE 802.15.4 and ANT) two to three times. The attempts to characterize BLE using the analytic methods were done by Gomez, Demirkol and Paradells in [11] and by Liu, Chen and Ma in [16]–[18]. In [11] the authors analyzed the effect of Bit Error Rates and connection parameters on throughput of a BLE P2P link. In [16]–[18] the authors modeled and analyzed the neighbor discovery procedure and suggested some strategies for enhancing the discovery performance.

Nonetheless, in general, the networking aspects of the BLE technology were not getting much attention up to now. Among the major reasons for this are: a) the complexity of the BLE communication, which impedes its analysis and b) the absence of the tools capable of simulating the BLE networks. In this paper we handle the second issue by introducing the novel simulation tool intended for simulating the communication in the networks of BLE devices.

## II. Bluetooth Low Energy Technology Overview

Similar to the classic Bluetooth, BLE's protocol stack consists of the two components: a BLE Controller and a BLE

**978-1-4799-4912-0/14/$31.00 © 2014 IEEE**

Host (see Fig. 1), which either reside on the same physical device or on the different devices. The Controller is the logical entity that is responsible for the physical (PHY) layer and the link layer (LL). The Host implements the functionalities of the upper layers which we leave out of the discussion.

On the physical layer (PHY) BLE uses the Gaussian Frequency Shift Keying (GFSK) modulation with a bandwidth bit period product equal to 0.5 and the symbol rate of 1 megasymbols per second. To simplify the implementation even further, the BLE transceivers use very short data packets with the maximum length of 47 bytes [4]. For a BLE transmitter, the power of output radio signal might range from -20 to +10 dBm and the sensitivity level of the BLE receiver is below -70 dBm [1]. Similar to the classic Bluetooth, BLE operates in the license-free industrial, scientific and medical (ISM) 2.4 GHz band, which is divided into 40 2-MHz wide channels. Three of those, which are located between typically used wireless local area network channels, are assigned specifically for advertising and discovery of services and are called *advertising channels*. The 37 *data channels* are intended for transferring the data between devices. The data transmission in BLE is bound to time units known as advertising and connection events.

The advertising events are used by the BLE transceivers for broadcasting small blocks of data or to agree on the parameters of the connection established in the data channels. At the beginning of an advertising event the advertiser, i.e., the device which has some data to transmit, sends an advertising frame. Using the different advertising frame types, the advertiser might either encapsulate up to 31 bytes of data directly in the advertisement frame or can indicate its capability to establish a connection in data channels. In the latter case, after transmitting a frame, the advertiser starts receiving and waits for possible connection establishment requests. If the connection request from a device (which is referred to as the initiator) is received, the two devices might start peer-to-peer connection in the data channels. Depending on the stack implementation and the application, a BLE advertiser might either send its advertisements in a single advertising channel or sequentially switch between different advertising channels.

Once a connection in data channels is established, the *initiator* becomes the *master* and the *advertiser* - the *slave*. Note, that BLE presumes, that a master is more resource-rich than a slave [2]. In the beginning of each connection event (referred to as the connection event anchor point) the used radio channel is changed following the pre-agreed sequence. The communication in each event is started when the master sends a frame to the slave. The master and the slave alternate sending the frames on the same channel while at least one of the devices has data or until the current connection event ends. 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 misses a radio packet. Once the connection event is closed, both master and slave might switch to a low-power sleep mode and wait until the start of the next connection event. The parameters of the connection (e.g., the connection event interval - *connInterval* or the list of used data channels) might be modified on the run. The connection is terminated by either device once the link is not required or automatically on connection supervision timeout (ranges from 0.1 to 32 s).

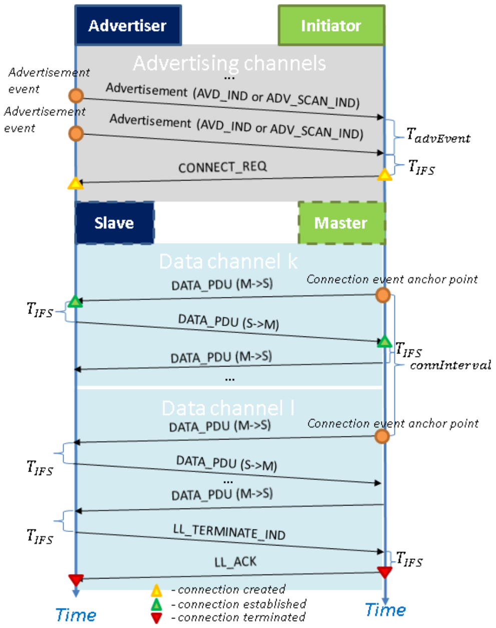

Fig. 2. Illustration of advertising, connection establishing, data transferring and connection terminating in BLE

The timing of connection events is determined through two parameters, namely the *connInterval*, and the slave latency (*connSlaveLatency*) [1]. The *connInterval* is a multiple of 1.25 ms ranging from 7.5 ms to 4.0 s. The *connSlaveLatency* ($connSlaveLatency \leq 500$) defines the maximum number of consecutive connection events in which a slave is not required to listen to the master and enables energy saving. The period between the frames on the same data channel equals to the Interframe Space period (IFS) set at 150 $\mu s$. The maximum LL payload of a BLE data frame is 27 bytes [4], [19].

The whole procedure of advertising, link establishing, data transferring and connection terminating is illustrated in Fig. 2.

## III. BLE Simulation Tool

### A. Overview

As discussed in Section II, BLE devices have various modes of operation and the protocol is rather complex, which makes it hard to use the analytical methods for studying BLE networks. Meanwhile, to the best of author's knowledge, no network simulation tools support BLE at a time. This motivated us to develop new BLE simulation tool for investigating the networking aspects of the protocol. Once the testing and documentation will be finalized, we plan to place the tool in the public domain. As the basis for the tool we used the MiXiM framework (v2.2.1) [20] based on the popular OMNeT++ engine (v4.2.2) [21]. In order to enable the multichannel communication, we extended the model of the PHY layer. The LL model was implemented as a state machine with five states (i.e., Standby, Advertising, Scanning, Initiating and Connection) as prescribed by the BLE specification v4.1 [1]. Although in the current version of the simulator the HCI and the host layers are not implemented and the packets are generated at the LL, this should not affect the accuracy of the communication simulations. All the parameters of the BLE communication (e.g., advertising and connection intervals, supervision timeout, frequency hop increment, the lists of the used data and advertisement channels) are defined in the simulation initialization file and remain constant during the simulation. The other restriction is the support of only one active link at a time for each simulated device. The security and the Ping command are not implemented at a time.

Even though we did our best to strictly follow the BLE specification,we had to make one exception. According to the specification (see pp. 2543-2544 in [1]),"the slave should listen for *windowWidening* before the start of the *slaveExpectedAnchorPoint* and until *windowWidening* after *slaveExpectedAnchorPoint* for the master's anchor point", where *windowWidening* is given by:

$$windowWidening = ((masterSCA + slaveSCA)/1000000) \cdot timeSinceLastAnchor \quad (1)$$

As one can see, for highly accurate clocks (i.e., low values of $masterSCA$ and $slaveSCA$) the $windowWidening$ is close to zero (e.g., if $masterSCA = slaveSCA = 10$, $timeSinceLastAnchor = 10ms$ then $windowWidening = 0.2\ \mu s$). If this occurs, the listen timer expires even before the slave might receive the first bit of the packet sent by master. To handle this issue, in our simulation model, we arm the listening timer $max(windowWidening, maxDataPacketDuration)$ after the *slaveExpectedAnchorPoint*, where $maxDataPacketDuration = 328\ \mu s$ is the maximum possible duration of a data-channel packet.

The developed tool enables one to simulate the operation of the BLE devices in various modes including: advertising in one or multiple advertising channels, scanning one or multiple advertising channels, establishing the connection, connection upkeep and terminating. The collected during the simulations data include, but are not limited to: the number of sent/received packets by each node, the state of each node and the used radio channel at each moment of time, consumed current and energy.

### *B. Evaluation of the Developed Simulator*

In order to ensure that the developed simulator provides feasible results we conducted a series of experiments and compared the results with the ones previously reported, and with the results of hardware measurements.

First, we checked the maximum throughput of a P2P BLE link. In [4] we have shown that the maximum LL throughput for a P2P link is 319.5 kbit/s. In the experiment we simulated a two-node network with one node acting as a master and the other one - as slave. On the master node we placed 269 973 bytes of data (i.e., 9 999 LL packets) and measured the time from connection establishment up to the last packet transfer. Using the maximum *connInterval* (i.e., 3200), the data transfer required 6.760086 s, which gives the throughput of 319.491 kbit/s. This correlates pretty well with the analytic result.

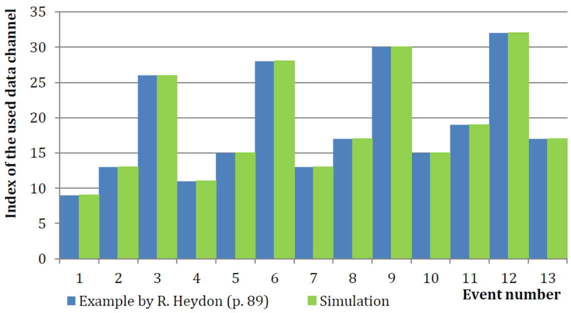

Fig. 3. Frequency hopping sequence for data channels (first 8 data channels are excluded, *hopIncrement=13*)

Second, we tested the frequency hopping. For this, the data channels with numbers 1..8 were excluded from use and *hopIncrement=13* and *connInterval=6* (i.e., 7.5 ms) were set. The sequence of channel transitions for a simulated node and the reference example (see p. 89 in [2]) are shown in Fig. 3.

Finally, we evaluated the results of energy consumption simulations. For this, we set the current consumption of nodes in the receive, transmit, sleep and switching states using the data from the datasheet of Texas Instruments (TI) CC2540 BLE transceiver [22]. Then, we measured the current consumption profile for transmitting the radio packet with maximum LL payload for the hardware CC2540 evaluation modules [23]. The measurements were executed using the current-shunt method on a 10.1 Ohm resistor using the oscilloscope. The current consumption profiles of the real-life hardware transceiver and of the simulated device are illustrated in Fig. 4. As one can see, the simulator accounts quite well for the consumption of the radio transceiver, but it does not take into consideration the consumption of the microcontroller for preparing the packet and processing the data.

From the presented evaluation results, one can see that the developed simulator is capable of simulating multiple aspects of BLE protocol with sufficient precision.

## IV. Results of BLE Network Simulation

First of all, we investigated the effect of the *connInterval* on the maximum P2P throughput. For this, we used a network consisting of one master and one slave node. The values of *connInterval* were varied from 2 to 3200. Note, that the minimum value for *connInterval* according to [1] is 6. The simulations were conducted for the two cases: when the data were transferred from master to slave and vise versa. The size of the data blocks for both cases was 26973 bytes. The results are presented in Fig. 5.

Fig. 5 reveals that for low *connInterval* the throughput of the link is subject to variations and the higher values of *connInterval* might result in the lower average throughput (compare *connInterval=6* and *connInterval=8*). Although this result might seem strange at first, it is quite natural. Each BLE

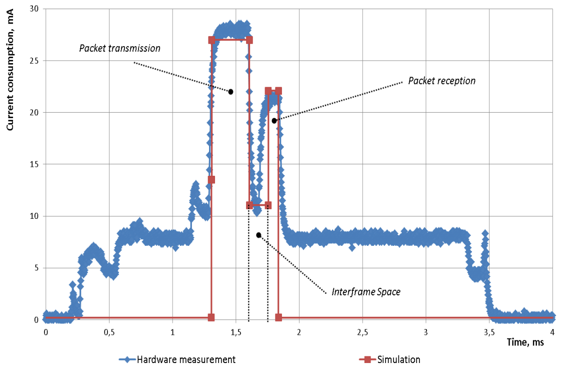

Fig. 4. Current consumption of a hardware BLE transceiver and of the simulated device while sending a LL packet with 27-byte payload and getting an acknowledgment

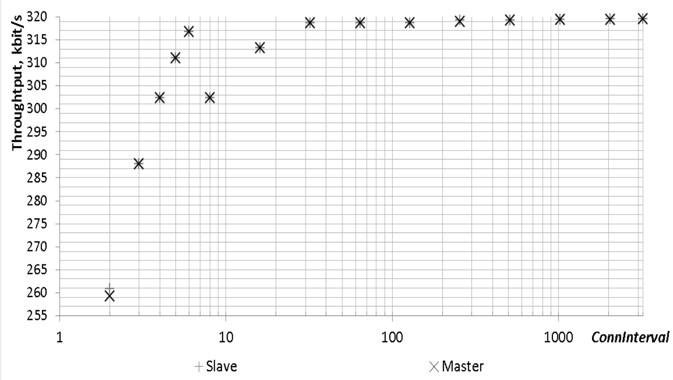

Fig. 5. Effect of *connInterval* on the throughput of BLE link at LL

connection event has pre-defined duration and starts with a packet transmitted by the master to the slave. Therefore, if the last packet in the event is sent by master (i.e., there is no time for a slave to send an acknowledgment), the very same packet is retransmitted in the beginning of the next connection event. This causes the degradation of the total throughput seen for *connInterval=8* in Fig. 5. With the increase of the *connInterval*, the significance of this effect diminishes and the LL throughput approaches 319.5 kbit/s. Note, that in the developed BLE LL implementation, when there is sufficient time before the end of connection event and a node has enough data, it always sends a packet with maximum payload. In the case, if the full-sized packet can not fit the remaining time, a node estimates how big packet it might send. If the payload of the resulting packet is $>$ 0, the node sends the packet, otherwise - closes this event.

Second, we simulated a network consisting of 1 to 64 pairs of BLE nodes placed in the area of 10 by 10 meters. As our major goal was to investigate the effect of having at the same time multiple active BLE links in the same region, for the simulations we assumed that each pair of nodes has the same amount of data to be transfered (i.e., 26973 bytes, stored in the buffer of each slave). All nodes started in the advertisement channels and established the connection (using ADV_DIRECT_IND events with *advInterval*=100 ms). Then the nodes switched to the data channels for transferring the data. If due to the interference from the other nodes the connection was not established in time or was canceled due to supervision timeout, the nodes returned to the advertising channels and re-established the connection. The experiment continued until all the slaves have transmitted the data to the respective masters. After finishing the data transfer, the nodes terminated the connection and entered sleep mode. During the experiments, the nodes used all 3 advertising channels and all 37 data channels. Note, that the decision on the packet reception by PHY layer was made based on signal to noise ratio, whilst the transmissions of the other nodes in the same channel was accounted as the additional noise. Therefore, in our simulations, multiple radio links might successfully operate on the same frequency channel at the same time in remote regions of the network. The results of the experiments revealing the time for transferring the data and the energy consumption of the slave nodes are presented in Figs. 6 and 7. For obtaining each point, the experiment was repeated 30 times for different network layouts. The average (marked 'x'), minimum and maximum values for these runs are depicted.

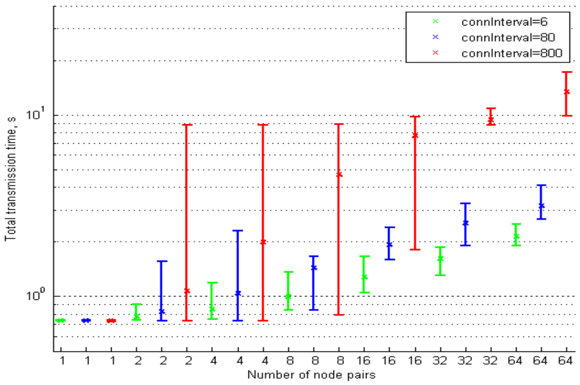

Fig. 6. Data transfer time for a multinode BLE network (*connInterval*=const, *hopIncrement*=uniform(5..16))

The presented results reveal that in the multi-node scenario the performance of BLE links varies significantly. Based on the extensive studies of the node's behavior, we identified two scenarios causing this.

First, let's consider the scenario when there are just two pairs of nodes: (M1,S1) and (M2,S2), where M denotes the master node and S - the slave. Assume that M1 starts the connection event in data channel $k$. Then, after time $\Delta t$, M2 switches to channel $k$ and starts its connection event. If the nodes are located so that the data transfer between M2 and S2 disturbs communication between M1 and S1, after the first packet loss (M1,S1) cancel the connection event. The problem arises if both pairs have the same *connInterval* and *hopIncrement*. In this case, the described situation repeats for *each* connection event and BLE does not have any mechanism to mitigate this. Of cause, if $\Delta t$ is lower than the time required

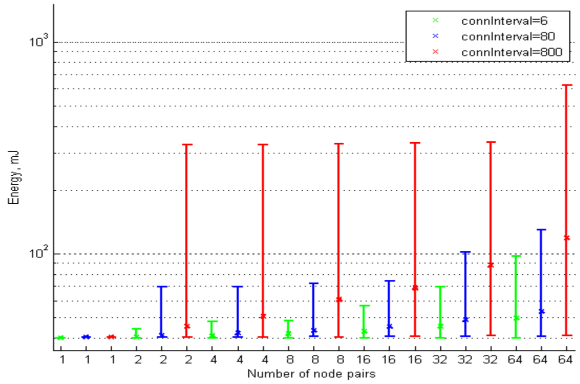

Fig. 7. Slave's energy consumption in a multinode BLE network (*connInterval*=const, *hopIncrement*=uniform(5..16))

to transfer the beacon packet and to get a reply from S1, then after supervision timer expiration (M1,S1) will terminate the connection. But if $\Delta t$ is higher than that, both M1 and S1 update the supervision timer and continue the connection. As easy to see, in the worst case (i.e., if $\Delta t$ is equal to the time for sending the beacon and getting the slave's reply), this will result in just a single data packet being transmitted by each (M1,S1) in each connection event.

The second issue is caused by the features of the connection establishment mechanism. As illustrated in Fig. 2, to establish the connection an initiator sends a connection request after which both nodes should switch to data channels. However, if the connection request packet is not received by the advertiser correctly, the latter continues advertising. Meanwhile, after sending the request, the initiator switches to data channels and starts sending the beacon packets. As the supervision timeout during the connection establishment is proportional to the desired *connInterval* (see p. 2539 in [1]), for high *connInterval* values, it takes significant time before initiator gets back in the advertising channels. Meanwhile, if $connInterval > advInterval$, the energy consumption of the resource-rich advertiser (i.e., the slave) during this time will be much higher than the one of the resource-poor initiator (i.e., the master).

## V. Conclusions

To the best of author's knowledge, the current paper is among the first ones focusing on the networking aspects of BLE protocol. In the paper we introduced the novel tool which enables BLE network-level simulations and discussed its capabilities and limitations. The presented results prove that the suggested simulator gives quite realistic results both in terms of throughput and energy consumption. Also, it was pointed out that the current version of BLE specification has an issue related to the definition of the slave's listening time in the beginning of a connection event which requires a workaround.

Using the presented simulator, we conducted the experiments which revealed the effect of the communication parameters on the BLE links performance in multi-node environment. The presented results pointed out two scenarios for which the features of the currently used in BLE packet loss handling and connection event cancellation mechanisms might cause significant degradation in the communication performance for the multi-device scenarios.

## References

[1] *Bluetooth Specification version 4.1*, The Bluetooth Special Interest Group, Kirkland, WA, USA, 2013.

[2] R. Heydon, *Bluetooth Low Energy: the developers handbook*. Upper Saddle River, NJ: Prentice Hall PTR, 2012.

[3] C. Gomez et.al.,"Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology," *Sensors*, vol. 12, no. 9, pp. 11734-11753, Aug. 2012.

[4] K. Mikhaylov et.al., "Performance analysis and comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI," *J. Sens. Actuator Networks*, vol. 2, no. 3, pp. 589-613, Aug. 2013.

[5] *Bluetooth Smart and Smart Ready products now available* (26.02.2014), Available: http://www.bluetooth.com/Pages/Bluetooth-Smart-Devices-List.aspx

[6] B. Yu et.al., "Bluetooth Low Energy (BLE) Based Mobile Electrocardiogram Monitoring System," *Proc. Int. Conf. Inf. and Automation*, Shenyang, China, 2012, pp. 763-767.

[7] M. Al et.al., "A Bluetooth Low Energy Implantable Glucose Monitoring System," *Proc. 8th European Radar Conf.*, Manchester, UK, 2011, pp. 377-380.

[8] A. J. Jara et.al., "Evaluation of Bluetooth Low Energy Capabilities for Continuous Data Transmission from a Wearable Electrocardiogram," *Proc. 6th Int. Conf. Innovative Mobile and Internet Services Ubiquitous Computing*, Palermo, Italy, 2012, pp. 912-917.

[9] Y. Park and H.Cho, "Transmission of ECG data with the patch-type ECG sensor system using Bluetooth Low Energy," *Proc. Int. Conf. ICT Convergence*, Jeju, ROK, 2013, pp. 289-294.

[10] K. Zidek, and J. Pitel, "Smart 3D pointing device based on MEMS sensor and Bluetooth Low Energy," *Proc. IEEE Symp. Computational Intelligence Control and Automation*, Singapore, Singapore, 2013, pp. 207-211.

[11] C. Gomez et.al., "Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link," *IEEE Commun. Lett.*, vol. 15, no.11, pp. 1187-1189, Sept. 2011.

[12] M. Siekkinen et.al., "How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee / 802.15.4," *Proc. IEEE Wireless Commun. and Networking Conf. Workshops*, Paris, France, 2012, pp. 232-237.

[13] M.C. Ekstrom et.al., "A Bluetooth Radio Energy Consumption Model for Low-Duty-Cycle Applications," *IEEE Trans. Instrum. Meas.*, vol.61, no.3, pp. 609-617, Mar. 2012.

[14] E. Mackensen et.al., "Bluetooth Low Energy (BLE) based wireless sensors," *Proc. IEEE SENSORS*, Taipei, Taiwan, 2012, pp. 1-4.

[15] A. Dementyev et.al., "Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario," *Proc. IEEE Int. Wireless Symp.*, Beijing, China, 2013, pp. 1-4.

[16] J. Liu et.al., "Modeling Neighbor Discovery in Bluetooth Low Energy Networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1439-1441, Sept. 2012.

[17] J. Liu et.al., "Energy Analysis of Device Discovery for Bluetooth Low Energy," *Proc. IEEE 78th Vehicular Technology Conf.*, Las-Vegas, NV, USA, 2013, pp.1-5.

[18] J. Liu et.al., "Modeling and performance analysis of device discovery in Bluetooth Low Energy networks," *Proc. IEEE Global Commun. Conf.*, Anaheim, CA, USA, 2012, pp.1538-1543.

[19] K. Mikhaylov and J. Tervonen, "Multihop Data Transfer Service for Bluetooth Low Energy," *Proc. 13th Int. Conf. ITS Telecommunications*, Tampere, Finland, 2013, pp. 319-324.

[20] *MiXiM project* (26.02.2014). Available: http://mixim.sourceforge.net/

[21] *OMNeT++* (26.02.2014). Available: http://www.omnetpp.org/

[22] *2.4-GHz Bluetooth low energy System-on-chip* (19.03.2014), SWRS084F, Texas Instruments, Available: http://www.ti.com/lit/ds/symlink/cc2540.pdf

[23] *CC2540 Evaluation Module Kit* (19.03.2014), Texas Instruments, Available:http://www.ti.com/tool/cc2540emk