Mechanisms for improving throughput and energy efficiency of Bluetooth Low Energy for multi node environment[1]

K. Mikhaylov* and T. Hänninen
Department of Communications Engineering, Centre for Wireless Communications, University of Oulu, Oulu, Finland.
*Corresponding author, Email: konstantin.mikhaylov@oulu.fi

Abstract. Bluetooth Low Energy (BLE) is a recently developed energy-efficient short-range radio communication technology, which is nowadays vastly becoming more and more popular. The technology aims to provide an energy-efficient low-cost wireless communication mechanism for a wide range of resource-limited consumer electronic devices and applications. In this paper we focus on the networking aspects of the technology. First, we overview fundamentals of BLE communication technology, discuss its capabilities and limitations, and compare BLE with state-of-the-art technologies. Furthermore, we identify and discuss two scenarios, for which standard BLE communication mechanisms show poor performance and propose two mechanisms capable of improving throughput and energy efficiency of BLE communication in multi node environment. The efficiency of the proposed mechanisms is proved via network level simulations, which are executed using the developed BLE simulation tool. The results of the simulations are reported and discussed. We believe that the presented results might be of some interest not only for academic researchers but also for practicing engineers.

Keywords: Bluetooth Low Energy, Bluetooth Smart, Network, Multi Node, Throughput, Energy, Improvement, Wireless

[1] *NOTE: the current manuscript is a pre-final version of the paper. The final version, including some minor modifications, has been published as:*
K. Mikhaylov and T. Hänninen, "Mechanisms for Improving Throughput and Energy Efficiency of Bluetooth Low Energy for Multi Node Environment", Journal of High Speed Networks Journal of High Speed Networks, Vol. 21, Issue 3, 2015, pp. 165-180.
and is available from the journal's web page from:
http://content.iospress.com/articles/journal-of-high-speed-networks/jhs518

1. Introduction

Bluetooth Low Energy (BLE) or Bluetooth Smart was standardized in 2010 as a part of Bluetooth Core Specification version 4.0 [3]. Development of the technology was motivated by a need for reducing the power consumption, complexity and price of radio transceivers. Despite carrying the name of its predecessor and operating in the same 2.4 GHz frequency band, BLE is an all-sufficient technology employing novel technical solutions, which make it incompatible with classic Bluetooth. The major target scenarios for BLE are different measurement and control applications, where battery-powered BLE sensors and actuators are interfaced with Personal Information Devices (PIDs), such as smartphones or laptops.

One of the major advantages of BLE compared to existing communication technologies is the compatibility with numerous commercial mobile phones, tablets and laptops. Bluetooth Special Interest Group (SIG) has defined two types of Bluetooth Smart compatible devices. The former ones, named "Bluetooth Smart Ready", are the PIDs. They encapsulate dual-mode Bluetooth transceiver supporting communication over both Bluetooth Classic and BLE. The latter ones, called "Bluetooth Smart", are BLE-only devices which can communicate between themselves and with Bluetooth Smart Ready devices. Between 2011 and 2014 many consumer electronics manufacturers replaced Bluetooth Classic radio transceivers with the chips supporting both Bluetooth Classic and BLE in their products. Today availability of BLE in PIDs provides a good starting point for developing a wide variety of human-oriented distributed measurement and control applications [43]. Therefore, we have witnessed an appearance of multiple Bluetooth Smart gadgets on the market over the last year and, undoubtedly, more are on the way. This makes BLE a very probable candidate for enabling the Internet of Things (IoT) concept and emphasizes the importance of studying the networking aspects of this technology.

In this paper we first discuss the results of the previous studies related to BLE technology in Section 2. In Section 3, we overview fundamentals of BLE communication, discuss the capabilities and limitations of the technology, and compare it with the state-of-the-art short range wireless communication technologies. In Section 4 we indicate and discuss in details two scenarios, in which standard BLE communication mechanisms show poor performance. The major contributions of the paper are two mechanisms targeting the discussed scenarios and aimed to improve throughput and energy efficiency of BLE communication in multi-node environment. They are introduced and discussed in Section 5. In Section 6 we present results of network simulations conducted using the developed network-level BLE simulation tool. The presented results prove feasibility and efficiency of the proposed mechanisms. Finally, Section 7 concludes the paper and summarizes the obtained results.

2. Related works

Although BLE standard was introduced a few years ago, the technology got serious attention from academic community quite recently.

The performances of BLE technology and of the commercial hardware BLE transceivers were investigated and compared with other state-of-the-art wireless communication technologies in [43] [9] [14] [48]. The reported results revealed that theoretical maximum Link Layer (LL) throughput for BLE is 319.5 kbit/s, which transforms into maximum application layer throughput of 236.7 kbit/s [43] [14] [35]. In terms of energy consumption, commercial BLE radios outperform e.g., IEEE 802.15.4 transceivers 2 to 4 times [43] [9] [48].

Over the recent years, a wide variety of novel applications has been developed employing wireless data transfer over BLE. The majority of reported applications are related to medicine and healthcare (e.g., [2] [7] [29] [30] [38] [39] [49] [53] [57]) and benefit from easy connectivity between a sensor and a smartphone. In addition, a few automotive ([21] [23] [12]) and consumer ([6] [58] [56] [55]) applications based on BLE technology have been reported. The feasibility of using energy-harvesting based BLE sensors in smart home environment was analysed and discussed in [45].

The novel designs of BLE transceivers and of particular transceiver components were reported by the authors in [54] [52] [33]. In [54] and [33] the authors proposed novel demodulator architectures for BLE transceivers; in [52] Wong et al. presented a low power transceiver chip supporting three different communication technologies including BLE.

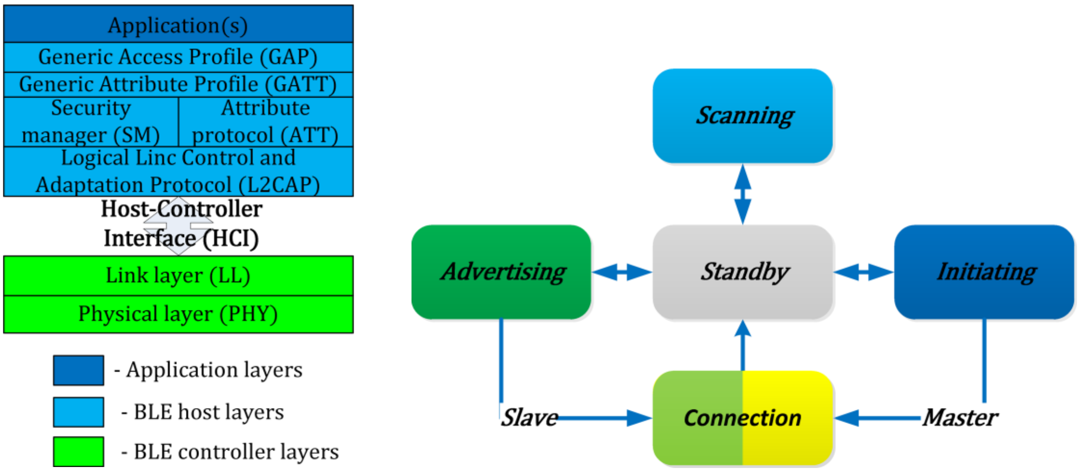
The new mechanisms for improving performance and capabilities of BLE communication were proposed in [35] [25] [27] [26] [41] [40] [36] [51]. Furthermore, a solution for transferring Internet Protocol (IP) packets over BLE was proposed in [35] and [36] . The first implementation of IP data transfer over BLE was reported in [51]. In [34] we proposed and reported implementation of the first multihop data transfer solution for BLE. The implementation of tree-based multihop BLE network solution was reported in [32]. The process of neighbour discovery for BLE was analysed in [25] and [27], and a new adaptive discovery mechanism was proposed in [26]. The security issues related to energy restricted wireless communication systems in general and the BLE protocol in particular were discussed in [4] [5] and [47] respectively.

Nevertheless, networking aspects of BLE technology have not received sufficient attention over the recent years. Among the major reasons causing this were: a) the complexity of BLE communication b) the absence of tools capable of simulating BLE networks. In [42] we reported the results of implementing a network-level simulator for BLE. In the paper we continue this work and target two scenarios, for which standard BLE mechanisms suffer from significant performance degradation in multi-node environment.

3. Fundamentals of BLE wireless communication protocol

Similar to classic Bluetooth, BLE protocol stack consists of two major components, namely BLE Controller and BLE Host (see Figure 1(a)). These two either reside on the same physical device or might be implemented by different devices. The Controller is a logical entity which is responsible for Physical (PHY) layer and LL. The Host implements functionalities of the upper layers. They include: L2CAP, GAP, ATT, GATT and SM. The Logical Link Control and Adaptation Protocol (L2CAP) multiplexes packets of the upper layers, manages connection establishment, configuration and destruction and might support packet segmentation. The Security Manager (SM) handles pairing, authentication, bounding and encryption of BLE communication. The ATTribute protocol (ATT) specifies mechanisms for discovering, reading and writing attributes on a peer device, and the Generic ATTribute profile (GATT) provides framework for discovering services and for reading and writing characteristic values. Finally, the Generic Access Profile (GAP) interfaces application and BLE stack. Host and controller are connected via the standartized Host Controller Interface (HCI).

In order to simplify the design and to minimize the cost of transceivers, BLE PHY is based on Gaussian Frequency Shift Keying (GFSK) modulation with a bandwidth bit period product equal to 0.5 and symbol rate of 1 mega-symbols per second. To simplify the implementation even further, BLE communication relies on



(a) BLE stack

(b)State diagram for BLE LL

Fig. 1. BLE technology fundamentals.

very short data packets with the maximum packet size of just 47 bytes. For a BLE transmitter, the power of output radio signal ranges from -20 to +10 dBm and the sensitivity level of BLE receiver should exceed -70 dBm. Similar to the classic Bluetooth, BLE operates in license-free Industrial, Scientific and Medical (ISM) 2.4 GHz band, which is divided into 40 2-MHz wide channels. Three of them, which are located between typically used wireless local area network channels, are assigned specifically for advertising and discovery of the services and are called *advertising channels*. The remaining 37 *data channels* are intended for transferring the data between BLE devices. The data transmission between BLE devices is bound to time units known as advertising and connection events.

The advertising events are used by BLE devices to broadcast small blocks of data or to agree on parameters of a connection to be established in data channels. In the beginning of an advertising event an advertiser, i.e. a device which either has some data to transmit or which expects to get some data, sends an advertising frame. Using different advertising frame types, the advertiser either encapsulates up to 31 data bytes directly in advertisement frame or announces its aptitude to establish a connection in data channels. In the latter case, after transmitting a frame, the advertiser switches to receive and waits for possible connection establishment requests. If the connection request from a device (which is referred to as an initiator) is received, the two devices start peer-to-peer connection in data channels. A BLE advertiser may either send its advertisements in a single advertising channel or sequentially switch between different advertising channels. The minimum time period between the starts of two consecutive advertising events according to BLE standard is defined as:

$$T_{advEvent} = advInterval + advDelay \tag{1}$$

where *advInterval* is an integer multiple of 0.625 ms in the range of 20 ms to 10.24 s, and *advDelay* is a pseudo-random value ranging from 0 ms to 10 ms.

Once a connection in data channels has been created, the initiator becomes master and the advertiser becomes slave (see Figure 1(b)). In the beginning of each connection event (referred to as the connection event anchor point) the used radio channel is changed following a pre-defined sequence. The communication in each connection event is started by a master, which sends a frame to a slave. The master and the slave alternate sending data frames on the same data channel while at least one of the devices has data to transmit or until the current connection event ends. In the case of either of the devices receives two consecutive frames with Cyclic Redundancy Check (CRC) errors, the

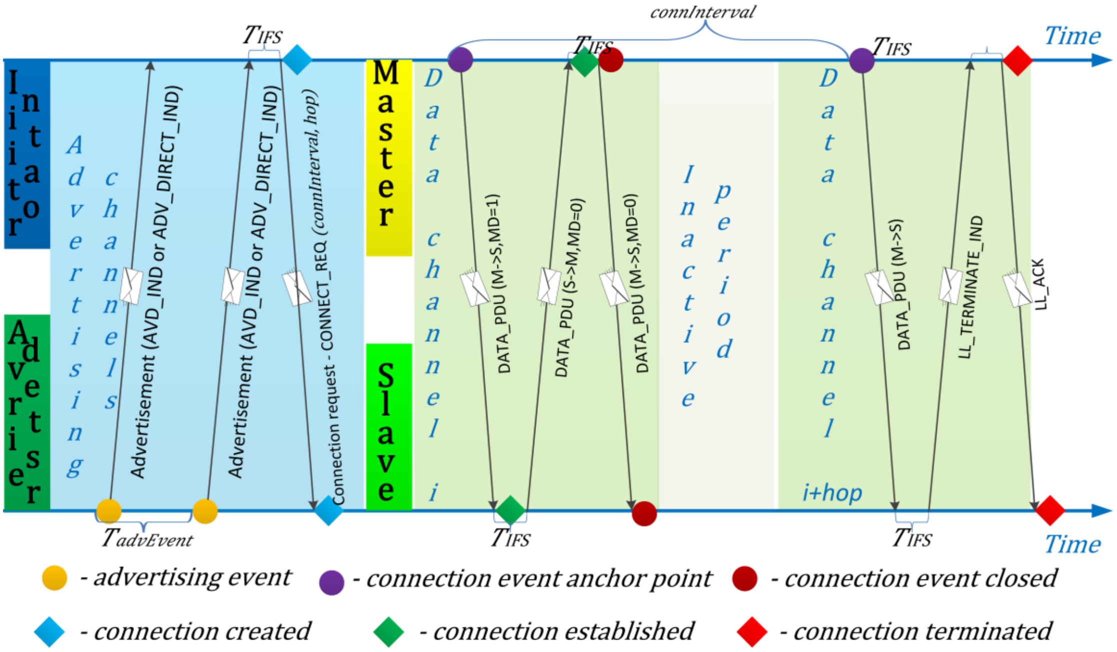


Fig. 2. Illustration of advertising, connection establishing, data transferring and connection terminating in BLE and related timing.

connection event is closed. The same happens if either device misses a radio packet. Once a connection event is closed, both master and slave might switch to low-power sleep mode until the start of the next connection event. The connection parameters (e.g., the interval between two connection event anchor points known as connection event interval - *connInterval*) may be updated on the fly without re-establishing the link. The connection is closed either by devices once the link is not required, or automatically due to connection loss timeout (so-called supervision timeout), which might range from 100 ms to 32 s. Note that BLE assumes that a master is typically more complex and richer on resources than a slave.

To mitigate the possible interferences caused by other systems, a set of the used data channels might be modified on the fly. The BLE features a mechanism enabling a master, which is aware of the spectrum situation, to exclude from use channels where strong interferences are expected. Note that the BLE specification does not define from where and how a BLE master should obtain information regarding the spectrum situation.

The timing for the connection events is determined using two parameters, namely the connection interval - *connInterval*, and the slave latency - *connSlaveLatency*. The *connInterval* is a multiple of 1.25 ms and ranges from 7.5 ms to 4.0 s. The *connSlaveLatency* defines the maximum number of consecutive connection events in which a slave is not required to listen for packets from a master. The period between two frames on the same data channel equals to the Interframe Space period (IFS) set at 150 µs. The whole procedure including advertising, connection establishment and data transferring for BLE is illustrated in Figure 2.

An important part of BLE stack is the client-server mechanisms enabling the discovery of available services, applications and providing the devices with the universal mechanism for data exchange. Those are implemented by ATT, GATT and GAP layers. The further details and the practical issues are discussed e.g., in [16].

The capabilities of BLE and of state-of-the-art and perspective wireless communication technologies are summarized in Table 1. Table 2 compares the characteristics of the commercial transceivers for Bluetooth, BLE, IEEE 802.15.4 and IEEE 802.11.

Table 1: Comparison of short-range wireless communication technologies (based on data from [3] [43] [9] [8] [1] [10] [11] [15] [17] [18] [19] [22] [24] [28] [37] [46] [50] [44]).

Protocol	Year[1]	Applications	Standardized layers[4]	Commercial devices	Frequency bands	Modulation	Channel spreading method	Symbol rate, kbit/s	Max. packet payload	Channel access method	Network topology	Multi-hoping support	Number of nodes
IEEE 802.15.4	2003	low-rate wireless personal area networks	PHY, DL(MAC)	available	868/915 MHz, 2.4 GHz	BPSK/ O-QPSK	DSSS	<250	127 bytes	CSMA/ CA	P2P, star, mesh, hybrid	supported	n/a
ISO 18000-7 (DASH7)	2004	ultra low energy wireless communication	PHY, DL	unavailable[5]	433 MHz	FSK/ GFSK	Not applicable	27.78[8]	249[8] bytes	slotted aloha	master-slave	limited support	n/a
IEEE 802.15.1 (Bluetooth)	1998	wireless personal area network	All layers	available	2.4 GHz	GFSK	FHSS	1000	2745 bits	FH-CDMA/ TDMA	master-slave	limited support	7 active per piconet
IEEE 802.11b/g	1999	wireless local-area-networks	PHY, DL (LLC + MAC)	available	2.4 GHz	DBPSK/ DQPSK	DSSS	11000/ 54000	2304 bytes	CSMA/ CA	P2P, star, mesh, hybrid	supported	unlimited
IEEE 802.15.4a	2007	wireless personal area networks with ranging capability	PHY, DL(MAC)	available	250-750 MHz, 3-5 GHz, 6-10 GHz	BPSK-BPPM	TH-IR UWB	27240	1209 symbols	CSMA/CA or slotted aloha	P2P/star	supported	<65536
IEEE 802.15.1-4 (Bluetooth Low Energy)	2010	low-power and low-latency short range wireless communication	All layers	available	2.4 GHz	GFSK	FHSS	1000	23 bytes	FH-CDMA/ TDMA	star	not supported	unlimited
ISO/IEC 14543-3-10 (EnOcean)	2012	wireless solutions with ultra-low power consumption and energy harvesting for building automation	PHY, DL, NWK	available	315/868 MHz	ASK	n/a[2]	125	not defined	CSMA/ CA	P2P, star, mesh, hybrid	supported	>65536
ITU G.9959 (Z-Wave)	2012	wireless communications protocol for home automation	PHY, DL(MAC)	available	868/915 MHz	FSK/ GFSK	n/a[2]	100	158 bytes	CSMA/ CA	P2P, star, mesh, hybrid	source routing	232 per network
IEEE 802.15.6	2012	wireless body area networks	PHY, DL(MAC)	unavailable	400/433 /868/915 MHz, 2.4 GHz	numerous[6]	Several options	<15600	255 bytes	CSMA/CA /slotted aloha/ TDMA	star	limited support (up to 2 hops)	64
Proper protocols	n/a[2]	n/a[2]	PHY, sometimes - DL(MAC) or NWK	available	400/433 /868/915 MHz, 2.4 GHz	mostly often: ASK, FSK, GFSK	Several options	usually <2000	usually <255	CSMA/CA /slotted aloha/ TDMA	n/a[2]	n/a[2]	n/a[2]
IEEE 802.11ah	2015[3]	sub 1 GHz sensor and smart metering network	PHY, DL(MAC)	-	868/915 MHz	Tbd[7]	Tbd[7]	Tbd[7]	Tbd[7]	Tbd[7]	Tbd[7]	Tbd[7]	Tbd[7]

[1] – year of initial specification release

[2] – not applicable

[3] – currently is under development, release is planned for 2015

[4] – PHY – physical, DL – data-link, MAC – media access control, LLC - logical link control, NWK – network

[5] – commercial radios compatible with DASH7 PHY are existing, but those do not include the appropriate DASH7 stack implementation

[6] – depending on the mode can be used: DBPSK, DQPSK, D8PSK, GMSK, OOK, CP-BFSK, DPSK, FM-UWB

[7] – to be defined

[8] – DASH7 Mode 2 (2009) supports the data rate of 250 kbit/s and packet payload of 249 bytes

The recent studies have shown that, in theory, a BLE link can provide 1.5-2 times higher throughput and is more energy efficient than IEEE 802.15.4 with 2450 DSSS physical layer. Another important benefit of BLE technology is the cost. As the Table 2 reveals, the commercial BLE transceiver chips are cheaper than the transceivers for competing technologies.

Among the major limitations of BLE technology are restriction of supported network topologies to only a single-hop scenario (i.e., point-to-point or star topologies), lower distance of communication compared to e.g., IEEE 802.15.4, and the use of the stop-and-wait flow control mechanism based on cumulative acknowledgments, which might

Table 2: Comparison of commercial transceivers operating in 2.4 GHz ISM band (based on data from [3] [43] [14] [10] and on analysis of currently available commercial transceivers).

Technology	**Modulation**	**Datarate,** *Mbit/s*	**Maximum TX Power,** *dBm*	**Minimum RX sensitivity,** *dBm*	**Transceiver price**^{2} **(normalized)**	**Current in TX**^{3}**,** *mA*	**Energy in TX**^{3} $\mu J/byte$	**Typical network topology**
IEEE 802.15.4 (2450 DSSS)	O-QPSK	0.25/ 0.192^1	10	-85	0.9	28	3.5	Mesh, P2P, Star
Bluetooth 1.2	GFSK	1/0.732^1	20	-70	0.7	44	1.44	Star, P2P
BLE	GFSK	1/0.320^1	10	-70	0.6	18	1.35	Star ,P2P
IEEE 802.11g	OFDM	54/31.4^1	20	-77	1	137	0.1	Star ,P2P

^{1}*-over-the-air/ maximum theoretical application throughput*

^{2}*-for low-end commercial transceiver chips on market*

^{3}*-transmit mode, 3V supply, 0 dBm TX power*

cause degradation of the protocol performance in the lossy environment [42].

4. Targeted scenarios

In our previous study [42] we presented a simulation tool, which extends the popular OMNeT++ network simulator by adding the PHY and LL models of BLE. Using this tool, we executed a series of experiments to study the network-level performance of BLE. The experiments enabled us to identify two scenarios causing degradation of link performance in multimode scenarios. The further details about the experiment set-up can be found from [42] and below we will discuss the detected problems and the preconditions for those.

4.1 Collision during connection establishment

The first problem is caused by the specifics of connection establishment mechanism in BLE. As depicted in Figure 2, to establish a connection initiator sends a connection request after which both nodes should switch to data channels and continue there. However, if a connection request packet is not received by advertiser correctly, the latter stays in advertising channels and continues advertising. Meanwhile, after sending a request initiator switches to data channels and starts sending beacon packets. Since the supervision timeout during connection establishment is proportional to the used $connInterval$, significant time might pass before an initiator closes unsucesful connection and returns to advertising channels. In addition, in the case if $connInterval > advInterval$, the energy consumption of potentially resource-poor advertiser (i.e., a slave) during this time will be much higher than the one of resource-rich initiator (i.e., a master). This scenario is illustrated in Figure 3.

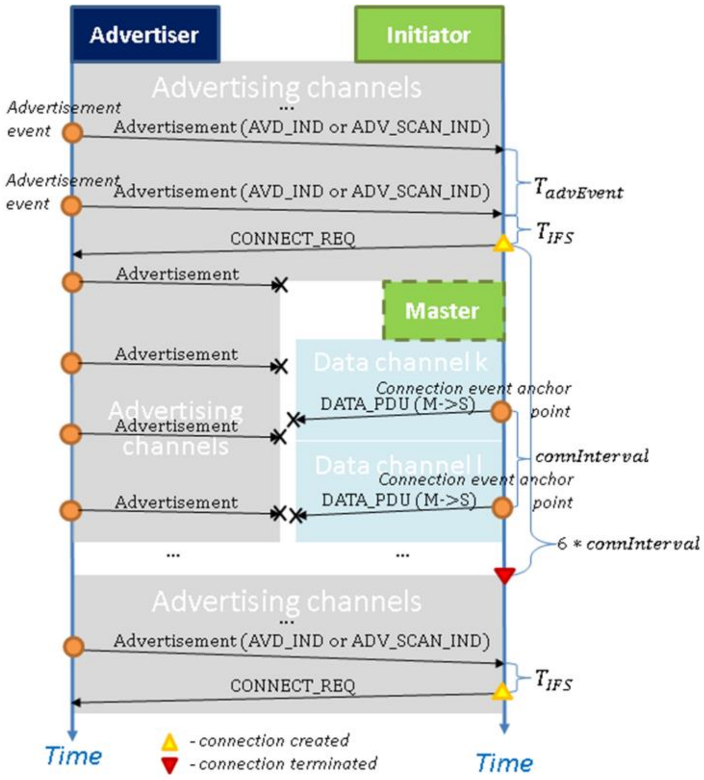


Fig. 3. Illustration of first detected scenario.

The discussed problem might arise if there are at least two nodes transmitting in the same advertising channel at the same time. Although this is not very likely to happen if all BLE devices in a network start advertising at random moments of time and do not return to advertising channels once a connection is successfully established, for a network with event-triggered data transfers the problem might arise much more often. E.g., consider the case of m BLE sensors and multiple gateways located in the same area. Assume, that the sensors do not transmit any data if the sensed parameter is below the specified threshold and issue an alarm once the sensed parameter exceeds the threshold. If all sensors detect an event and start advertising at the same time, have the same *advInterval* and start on the same channel (which is quite likely given that the sensors are identical and serve one application), the probability of having at least one collision during connection establishment phase is given by:

$$P_{collision}(m) = \begin{cases} 0, m = 1 \\ 1, 1/(m-1) \geq T_{ADV}/T_{rand} \\ 1 - \left(1 - \dfrac{(m-1)T_{ADV}}{10ms}\right)^{m}, else \end{cases} \qquad (2)$$

where $T_{ADV} = 630 + 8 \cdot n$ μs is the time required for establishing BLE connection (i.e., total time for sending advertising packet, IFS, and receiving the connection request (CONNECT_REQ); n stands for the payload of ADV packet; $n \in [0,31]$ for ADV_IND or n=6 for ADV_DIRECT_IND), $T_{rand} = \max(advDelay) + T_{ADV}$.

The probability of collisions during connection establishment phase as a function of the number of BLE sensors is illustrated in Figure 4. Although not each collision will cause a loss of connection establishment packet resulting in the described problem, this outcome is still quite possible.

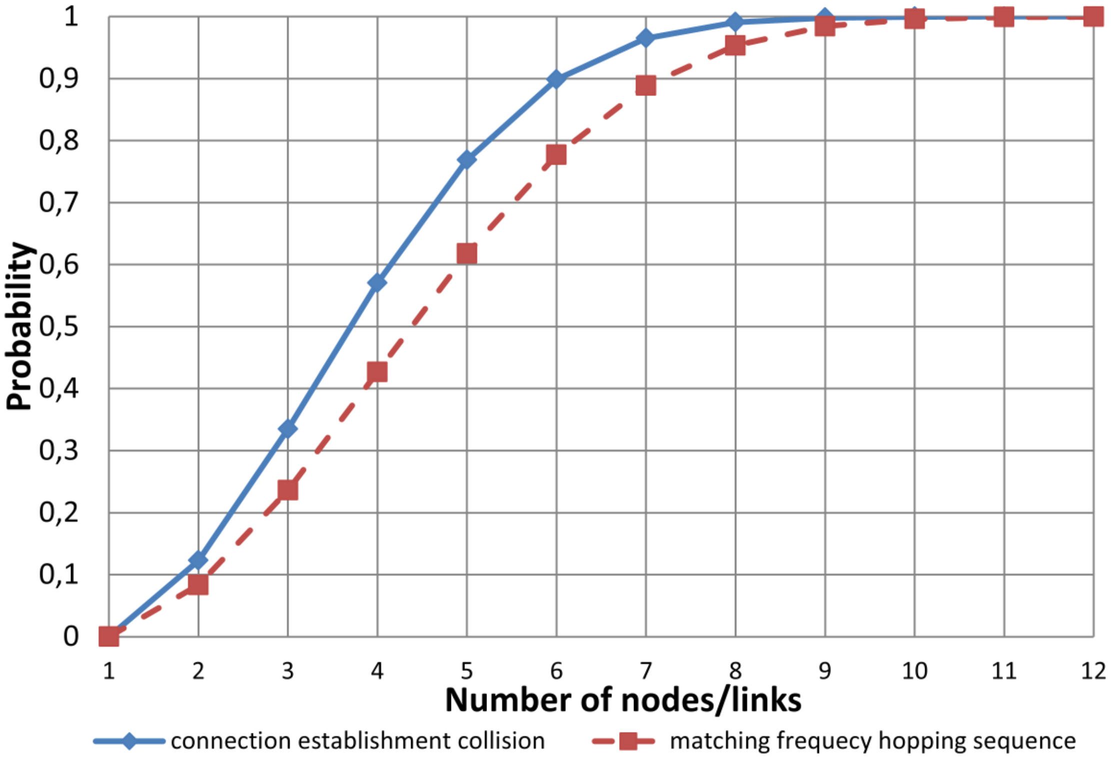


Fig. 4. Effect of number of nodes on the probability of discussed scenarios coming to life.

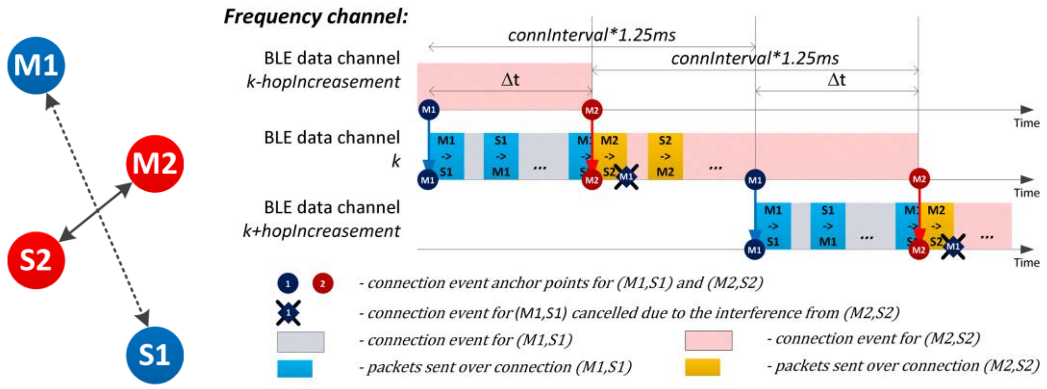
Permanent connection overlap

The other problem occurs more rarely, but might have a really devastating effect. E.g., consider the two pairs of nodes, namely (M1,S1) and (M2,S2) (where M denotes a master node and S - a slave) depicted in Figure 5. Assume that *M1* starts a connection event in data channel *k*. After time period Δt, *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 the communication between *M1* and *S1*, after the first packet loss pair *(M1,S1)* stops the communication and cancels the connection event. The problem arises if both pairs use the same *connInterval* and *hopIncreasement*. In this case the described situation repeats for *each* connection event and BLE does not have any mechanism to mitigate this. Note, that the channel hopping sequence for BLE is specified by the *hopIncreasement* parameter, which has only 12 valid values (for each connection a random value in the range of 5 to 16 is set [3]). The similar problem also arises in the case if a period of channel hopping sequence repeat plus one *connInterval* of one node pair is close to the *connInterval* of another pair.

Given that in the target region there are *m* active links with the same *connInterval*, the probability of at least two links having exactly the same frequency hopping sequence is given by:

$$P_{sequence\_match}(m) = \begin{cases} 0, m = 1 \\ 1, m \geq 12 \\ 1 - \dfrac{12!}{(12-m)!\,12^m}, else \end{cases} \quad (3)$$

The effect of the number of nodes on probability of frequency hopping sequence match is illustrated in Figure 4. Similarly to the scenario discussed in Section 4.1, the match of connection hopping sequence is not the only precondition resulting in the described scenario. Moreover, the effect of this scenario is affected by the amount of the data each of the pairs transmits and the delay between starts of connection events for


M1
M2
S2
S1
Frequency channel:
BLE data channel
k-hopIncreasement
BLE data channel
k
BLE data channel
k+hopIncreasement
connInterval*1.25ms
connInterval*1.25ms
Δt
Δt
Time
Time
Time
M1 -> S1
S1 -> M1
M2 -> S2
S2 -> M2
...
- connection event anchor points for (M1,S1) and (M2,S2)
- connection event for (M1,S1) cancelled due to the interference from (M2,S2)
- connection event for (M1,S1)
- connection event for (M2,S2)
- packets sent over connection (M1,S1)
- packets sent over connection (M2,S2)

(a)Example network layout

(b) Illustration of the discussed scenario

Fig. 5: Consequence of the mutual interferences for the two BLE links having same *connInterval* and *hopIncreasement*.

```
procedure ACE(connInterval_desired)
  send CONNECT_REQ with connInterval=6..10;
  switch to next data channel & initialize supervision timeout;
  while !timeout loop
    if(connection event start)
      send first packet;
    end if;
        if(reply received)
      send LL_CONECTION_UPDATE_REQ with connInterval=connInterval_desired;
       return true;
      end if;
   end loop;
   return false;
 end procedure
```

Fig. 6: Pseudocode of Accelerated Connection Establishment (ACE) mechanism.

different links Δt. Nonetheless, as we have shown in [42], in the worst case the discussed problem results in just a single packet being transmitted over a BLE link in each connection event. This results in reduction of throughput of an affected BLE link to a mere 2.16 kbit/s given that *connInterval*=0.1s or 54 bit/s if *connInterval*=4s.

5. Proposed mechanisms

5.1 Accelerated Connection Establishment (ACE) mechanism

To handle the problem described in Section 4.1 we propose the mechanism named Accelerated Connection Establishment (ACE). The idea of this solution was proposed in [41] and below we extend it and discuss its implementation in more details.

The ACE is a double-stage connection establishment mechanism which uses different values for *connInterval* on each stage. Instead of setting the desired *connInterval* in the CONNECT_REQ packet as this is typically done (see Figures 2 and 3), we propose to use very small values of *connInterval* during the initialization phase. On one hand, in the case of CONNECT_REQ loss, this enables to reduce the connection supervision timeout (which equals $6 \cdot connInterval$ [3]) for the initiator and thus enables it to get back in the advertising channels much faster. On the other hand, if a connection is established successfully (i.e., if a master gets successfully a packet from a slave in data channels), the master might invoke BLE connection parameter update mechanism for changing the $connInterval$ to the desired value. The pseudocode describing ACE is depicted in Figure 6.

For a real-life BLE transceiver ACE can be implemented in two ways: either at the LL of the Controller or at L2CAP of the Host. In the former case instead of using *connInterval* specified by Host in HCI LE_Create_Connection command, the LL issues CONNECT_REQ with a small $connInterval$. Once the first data frame has been successfully received from the slave, the LL of the master issues LL_CONECTION_UPDATE_REQ and modifies *connInterval* to the one which was originally requested by Host in LE_Create_Connection command. Once the new *connInterval* is set the LL informs the Host about successful connection establishment using LE_Connection_Complete_Event message.

In the latter case Host on master device issues LE_Create_Connection command with small *connInterval* already set. Once getting the LE_Connection_Complete_Event message from LL (which is issued once master has sent the CONNECT_REQ packet) Host issues LE_Connection_Update command and modifies *connInterval* to match the desired value. Note, that according to BLE specification [3], the master should allow a minimum of 6 connection events during which a slave will be active before starting to use the updated parameters. Therefore, in case if a CONNECT_REQ was missed by advertiser the connection supervision timer of master always fires earlier than the new *connInterval* is taken in use.

Regardless of the chosen implementation the overhead for proposed mechanism compared to single-staged connection establishment procedure is just one packet - a LL_CONECTION_UPDATE_REQ of 22 bytes, transmission of which takes mere 176 μs.

5.2 *Link Collision Monitoring and Update (LCMU) mechanism*

To mitigate the second described problem we propose the Link Collision Monitoring and Update (LCMU) mechanism. The proposed mechanism is integrated into the LL of BLE stack and implemented as a procedure, which is launched each time a connection event is closed early due to packet loss or reception of packets with errors. Once this happens the procedure estimates the duration of active phase of connection event and compares it with the *ThresholdPhase* threshold. If the threshold is exceeded duration of active phase is stored in a data array. Once the number of elements in the array exceeds pre-defined number *NumPoints*, the procedure goes through all of them, calculates standard deviation and compares it with the *ThresholdDeviation*. If the threshold is exceeded the mechanism initiates a change of connection parameters. Namely, we propose leaving the same *hopIncreasement* and set $transmitWindowOffset = (ConstOffset + random(0, RandOffset)) \cdot connInterval$. This results in start time of the connection event being delayed by *ConstOffset* to *ConstOffset+RandOffset* $connInterval$ periods. The operation of LCMU can be tuned by modifying values of the two thresholds, as well as *NumPoints*, *ConstOffset* and *RandOffset*. The pseudocode describing LCMU is depicted in Figure 7.

```
procedure LCM(NumPoints,ThreesholdPhase,ThresholdDeviation,ConstOffset,RandOffset)
  if(connection closed early)
    T=time since start of connection event/connInterval;
    if(T<ThreesholdPhase)
      insert T in PointArray;
      if(size of PointArray>NumPoints-1)
        if(deviation of points in PointArray< ThresholdDeviation)
          send LL_CONECTION_UPDATE_REQ with
                    transmitWindowOffset=(ConstOffset+random(0,RandOffset))·connInterval
        end if;
      end if;
    else
      clear PointArray;
    end if;
  else
    clear PointArray;
  end if;
end procedure
```

Fig. 7: Pseudocode of Link Collision Monitoring and Update (LCMU) mechanism.

6. Evaluation

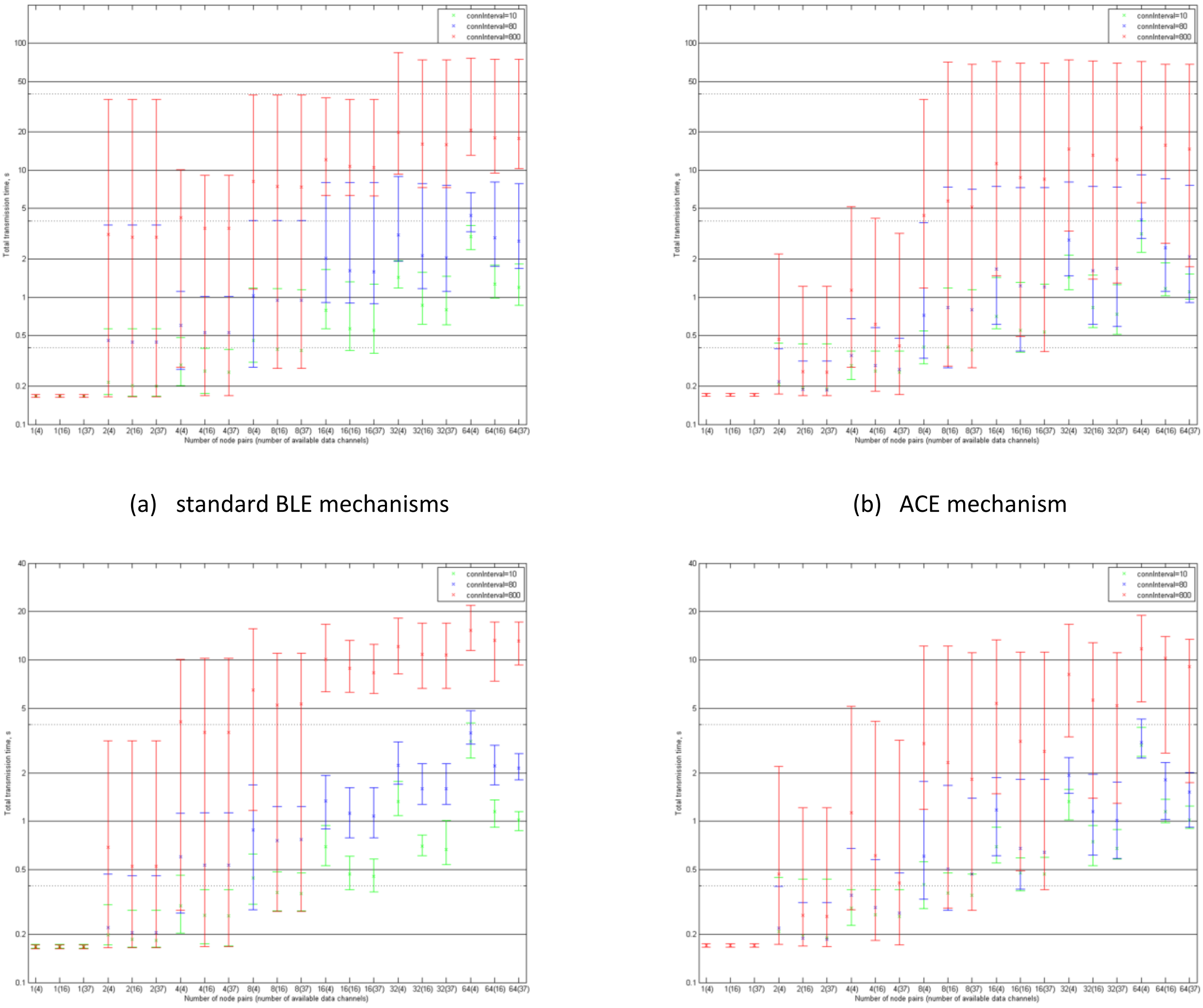
6.1 Simulation set-up

To validate the two proposed mechanisms we have integrated them with the BLE simulation environment, which has been presented in [42]. The simulator implements PHY, LL and simplified network layer and enables studying the communication and networking of BLE devices. The capabilities and limitations of the tool are discussed in [42] and [41]. The proposed mechanisms have been implemented in C++ code and integrated with the LL (LCMU mechanism) and network layer (ACE mechanism) of BLE stack.

For evaluating performance of the proposed mechanisms we simulated BLE networks consisting of 1 to 64 pairs of BLE nodes randomly placed over the area of 10 by 10 meters. In the beginning of simulation each master got 2000 bytes of data, which should be transferred to a slave. All the nodes started at the same moment of time in the advertisement channels and established the connection (using ADV_DIRECT_IND advertisement frames with *advInterval*=100 *ms*). Then the nodes switched to data channels. If the connection was not established properly (i.e., CONNECT_REQ was lost) or was broken before all the data had been transferred the nodes returned to advertising channels and re-established connection. Once all the data has been transferred the nodes terminated connection and switched to sleep mode. The experiment ended once all the nodes had completed data transfer. All the node pairs used the same *connInterval* and random *hopIncreasement* as defined in [3].

Note, that the chosen scenario matches few real-life situations. The first example is a smart home gateway based [45] security system, in which multiple camera nodes start streaming audio/video data once alarm is triggered. The other example is a wireless vehicle control system [21] [12] where large amounts of diagnostic data might be transferred immediately after a car start-up or after a traffic incident. Finally, the targeted scenario emulates a situation taking place after a disappearance of a strong wide-band interference, when multiple BLE nodes are attempting to re-establish connection and transfer their data.

In our study we simulated four different scenarios. In the first one all the nodes used the standard BLE mechanisms. In the second one all the nodes used ACE. The *connInterval* during first phase of ACE was set equal 10 ms. In the third scenario all the nodes monitored established links and modified connection parameters using the proposed LCMU with the following parameters: *NumPoints=3*, *ThreesholdPhase=0.5*, *ThresholdDeviation=0.1*, *ConstOffset=1* and *RandOffset=1*. Finally, we also tested the performance for the case when nodes had both ACE and LCMU active (labelled ACE+LCMU). The experiments were done for three different *connInterval* values, namely 10 (i.e,12.5 ms), 80 (i.e., 100 ms) and 800 (i.e., 1 s), and for different number of available data channels (4, 16 and 37). For each of the 6300 simulation runs, we monitored and logged the total time required for transferring the data and the energy consumed by each node. To minimize the influence of network topology, we repeated the experiment 25 times for each set of parameter with different network layouts. The minimum, maximum and mean values of the measured parameters were calculated. The obtained results are summarized in Figures 8 and 9.



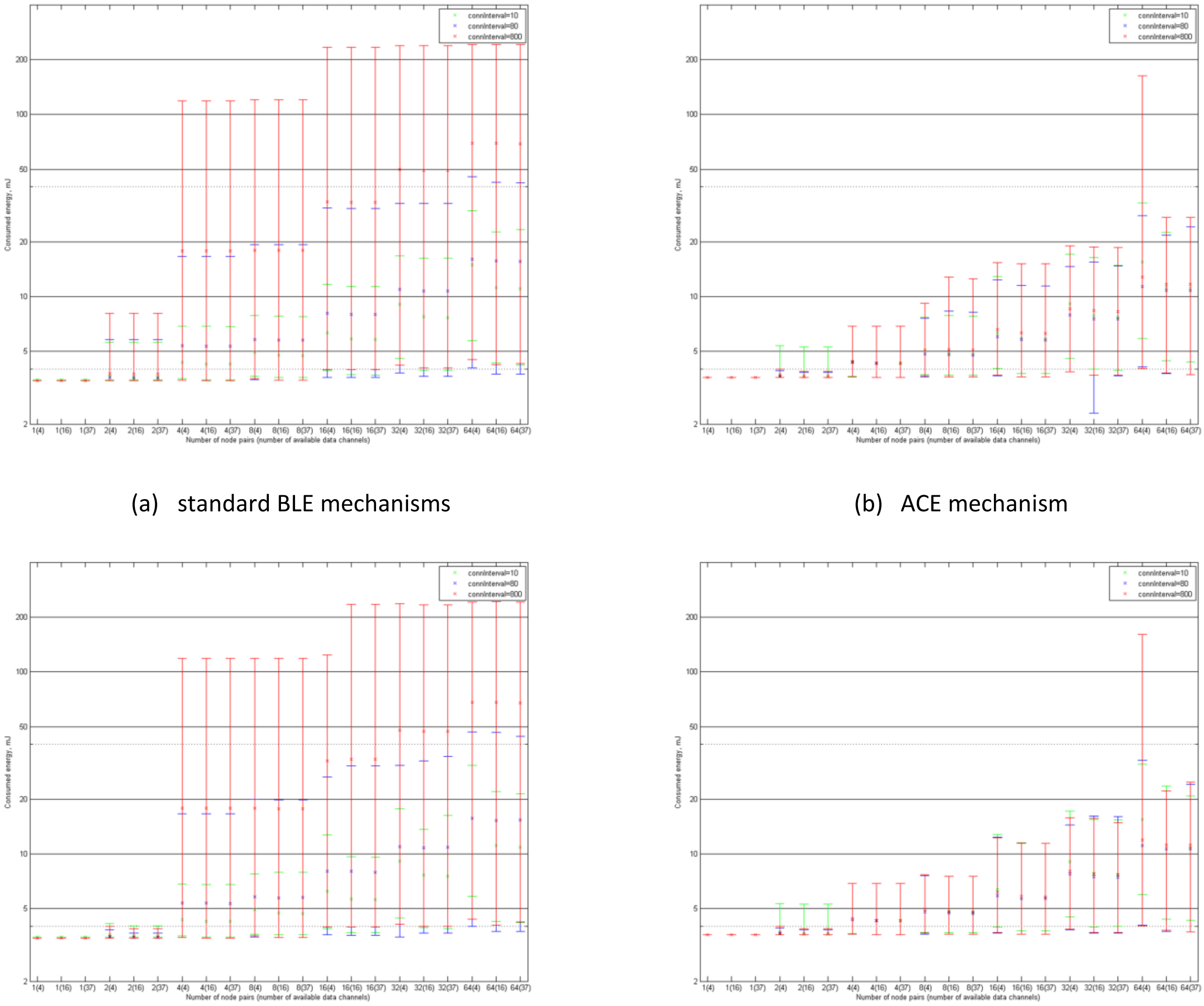
(a) standard BLE mechanisms

(b) ACE mechanism

(c) LCMU mechanism

(d) ACE + LCMU mechanisms

Fig. 8: Data transfer time.



(a) standard BLE mechanisms

(b) ACE mechanism

(c) LCMU mechanism

(d) ACE + LCMU mechanisms

Fig. 9: Slaves' energy consumption.

6.2 *Analysis of results and discussion*

Figure 8 illustrates the time required to complete the data transfers for all scenarios and runs. The mean value is marked with 'x', the minimum and maximum values are shown as well. Comparing Figures 8(b) and 8(a), one can easily see that use of ACE reduces average time of data transfer for multi-node scenarios. When the number of nodes is below 8, use of ACE can greatly reduce data transfer time. E.g., for 4 pairs of nodes (16 active channels and *connInterval*=100 ms) the average transmission time is reduced by 45%. For a high number of nodes and a small number of active data channels use of ACE gives minor benefit. E.g., for 64 pairs of nodes (37 active channels and *connInterval*=1000 ms) the total transmission time is reduced by 17%. From Figure 8(c) one can see that the use of LCMU dramatically reduces the maximum and average time of data transfer in multi-node environment. E.g., for 64 pairs of nodes (37 active channels and *connInterval*=1000 ms) the average time is reduced by 26% and the maximum time decreases from 74 s to 17 s. The combination of both mechanisms in a single solution (see Figure 8(d)) enables to improve the performance even further. E.g., for dense network (64 pairs of nodes, 37 active channels and *connInterval*=1000 ms) the combined solution outperforms the standard BLE mechanisms in terms of average time required for data transfer by 48%, ACE – by 37% and LCMU – by 30%.

The energy consumption of simulated slave nodes is illustrated in Figure 9. The comparison of Figures 9(a) and 9(b) shows that ACE reduces energy consumption dramatically. E.g., for 64 pairs of nodes (16 data channels and *connInterval*=1000 ms) nodes with ACE feature five times lower average energy consumption than the nodes using basic BLE connection establishment mechanism. The major reason for this is the reduction of consumption for idle advertising once an initiator has switched to data channels, which happens after CONNECT_REQ loss. Note that the benefit provided by ACE strongly depends from *advInterval/connInterval* ratio and if *advInterval > 6 connInterval* use of ACE will hardly be effective. Meanwhile, as one can see from Figure 9(c), although LCMU reduces total time for data transfer greatly, the improvement of energy consumption provided by this mechanism is minor. This happens due to fast detection of packet collision and switching to sleep featured by stop-and-wait flow control mechanism based on cumulative acknowledgments available in BLE. E.g., for 64 pairs of nodes LCMU reduces the average energy consumption compared to basic BLE mechanism by 0.5-2.5%. The combination of ACE and LCMU for dense networks provides the most efficient solution and has energy consumption 1.5-4% lower than the one of nodes with ACE.

Note that for the case if there are no interferences (i.e., only a single pair of nodes is present) the ACE-enabled nodes (as well as nodes using ACE and LCMU) in the simulated scenario consume 2-4% more energy and require 1-2% more time to complete data transfer than the nodes having only basic BLE mechanisms. This is caused by the need of sending an extra LL_CONECTION_UPDATE_REQ packet. Nonetheless, we do not consider this to be a significant drawback considering benefits provided by ACE in multi-node case.

7. Conclusion

To the best of our knowledge, the current paper is one of the first studies focused on the network aspects of BLE technology. In the paper we discussed in details capabilities of BLE technology and overviewed the specifics of BLE communication mechanisms. Then we identified two scenarios, in which the interference between BLE nodes results in significant increase of the time and energy required for data transfers. In this paper we first described the scenarios and then analysed conditions for those coming to life. Finally, we proposed two mechanisms which enable solving the detected problems.

The first mechanism, named Accelerated Connection Establishment (ACE), modifies the connection establishment procedure by dividing it into two phases with different values of connection interval. This solution reduces the time during which an initiator stays in data channels before returning to advertising channels if a connection establishment request is lost. The ACE can be implemented either by link layer of the Controller or by L2CAP of the Host on BLE transceiver.

The second mechanism, named Link Collision Monitoring and Update (LCMU), extends link layer of BLE with a solution which monitors active BLE links and detects if parameters of a link match the ones of links between neighbouring nodes. If such a match is detected, the parameters of a link are updated for improving the performance of data transfer.

Despite their simplicity, the proposed mechanisms appeared to be quite effective. The network-level simulations of multi-node BLE communication have shown that in a dense network with high connection intervals, the proposed mechanisms might reduce the time and energy for transferring the data four to six times. Nonetheless, we are certain that the efficiency of the proposed mechanisms might be improved even further by optimizing parameters of the mechanisms. We plan to investigate this in the future along with conducting other studies of issues related to networking of BLE devices.

We believe that the results presented in this paper might be interesting for academic researchers and practicing engineers and can be used for improving the performance of a wide range of BLE devices and applications.

References

[1] Z. Ahmadian and L. Lampe, Performance Analysis of the IEEE 802.15.4a UWB System, *IEEE Trans. Commun.* **57** (2009), 1474-1485.

[2] B. Allen et al., "Evaluation of fall risk for post-stroke patients using bluetooth low-energy wireless sensor," in *Proc. IEEE Global Commun. Conf.*, Atlanta, GA, USA, 2013, pp. 2598-2603.

[3] Bluetooth SIG, "Bluetooth Specification Version 4," Kirkland, WA, USA, 2010.

[4] A. Castiglione, A. De Santis, A. Castiglione, F. Palmieri, and U., Fiore, "An Energy-Aware Framework for Reliable and Secure End-to-End Ubiquitous Data Communications," in *5th Int. Conf. Intelligent Networking Collaborative Syst.*, Xi'an, China, 2013, pp. 157-165.

[5] A. Castiglione, F. Palmieri, U. Fiore, A Castiglione, and A. De Santisa, Modeling energy-efficient secure communications in multi-mode wireless mobile devices, *J. Comput. Syst. Sci.*(2014), in press.

[6] Young Seek Cho, Jaerock Kwon, Seyeong Choi, and Dae-Hee Park, "Development of smart LED lighting system using multi-sensor module and bluetooth low energy technology," in *Proc. 11th Annu. IEEE Int. Conf. Sensing, Communication, and Networking (SECON)*, Singapore, Singapore, 2014, pp. 191-193.

[7] D. Craven, M. Glavin, L. Kilmartin, and E. Jones, "Potential for extended battery life in mobile healthcare with Bluetooth low energy and signal compression," in *Proc. IET Irish Signals Syst. Conf.*, Maynooth, Ireland, 2012, pp. 1-6.

[8] L. De Nardis and M.-G. Di Benedetto, "Overview of the IEEE 802.15.4/4a standards for low data rate Wireless Personal Data Networks," in *Proc. 4th Workshop Positioning Navigation Communication*, Hannover, Germany, 2007, pp. 285-289.

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

[10] E. Ferro and F. Potorti, Bluetooth and Wi-Fi wireless protocols: A survey and a comparison, *IEEE Wireless Commun.* **12** (2005), 12-16.

[11] H. Fornazier, A. Martin, and S. Messner, "Wireless Communication: Wi-Fi, Bluetooth, IEEE 802.15.4, DASH7," 2012. [Online]. http://rose.eu.org/2012/wp-content/uploads/2012/03/Wireless-communication.pdf

[12] R. Frank, W. Bronzi, G. Castignani, and T. Engel, "Bluetooth Low Energy: An Alternative Technology for VANET Applications," in *Proc. 11th Annu. Conf. Wireless On-demand Netw. Syst. Services*, Obergurgl, Austria, 2014, pp. 104-107.

[13] C. Gomez, I. Demirkol, and J. Paradells, Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link, *IEEE Commun. Lett.* **15** (2011), 1187-1189.

[14] C. Gomez, J. Oller, and Paradells J., Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology, *Sensors* **12** (2012), 11734-11753.

[15] F. Haroon, K. M. Ahmed, and H. Rasheed, "An adaption of IEEE 802.15.4a in an industrial Field Bus," in *8th IEEE Int. Conf. Industrial Informatics*, Osaka, Japan, 2010, pp. 531-536.

[16] R. Heydon, *Bluetooth Low Energy: the developer's handbook*. Upper Saddle River, NJ: Prentice Hall PTR, 2012.

[17] ISO/IEC, "Parameters for active air interface communications at 433 MHz," 2004.

[18] ISO/IEC, "Wireless Short-Packet (WSP) protocol optimized for energy harvesting — Architecture and lower layer protocols," 2011.

[19] ITU, "Short range narrow-band digital radiocommunication transceivers – PHY and MAC layer specifications," 2012.

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

[21] A. Kandhalu, A.E. Xhafa, and S. Hosur, "Towards bounded-latency Bluetooth Low Energy for in-vehicle network cable replacement," in *Proc. Int. Conf. Connected Vehicles Expo*, Las Vegas, NV, USA, 2013, pp. 635-640.

[22] E. Karapistoli, F.-N. Pavlidou, I. Gragopoulos, and I. Tsetsinas, An overview of the IEEE 802.15.4a Standard, *IEEE Commun. Mag.* **48** (2010), 47-53.

[23] Xia Kun, Xu Xinyue, and Wang Nan, "Design of Vehicle Control System Based on Bluetooth Low Energy Smartphone Platform," in *Proc. Int. Conf. Electrical Machines Syst.*, Busan, ROK, 2013, pp. 1498-1501.

[24] K. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15.6 standard," in *Proc. 3rd Int. Symp. Applied Sciences Biomedical Commun. Tech.*, Rome, Italy, 2010, pp. 1-6.

[25] J. Lee, C. Canfeng, and M. Yan, Modeling Neighbor Discovery in Bluetooth Low Energy Networks, *IEEE Commun. Lett.* **16** (2012), 1439-1441.

[26] J. Lee, C. Canfeng, M. Yan, and Y. Xu, "Adaptive Device Discovery in Bluetooth Low Energy Networks," in *Proc. IEEE 77th Vehicular Technology Conf.*, Dresden, Germany, 2013, pp. 1-5.

[27] J. Lee, C. Canfeng, M. Yan, and Y. Xu, "Energy Analysis of Device Discovery for Bluetooth Low Energy," in *Proc. IEEE 78th Vehicular Technology Conf.*, Las-Vegas, NV, USA, 2013, pp. 1-5.

[28] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *33rd Annu. Conf. IEEE Industrial Electron. Society*, Taipei, Taiwan, 2007, pp. 46–51.

[29] Zhe-Min Lin, Cheng-Hung Chang, Nai-Kuan Chou, and Yuan-Hsiang Lin, "Bluetooth Low Energy (BLE) Based Blood Pressure," in *Proc. Int. Conf. Intelligent Green Building Smart Grid*, Taipei, Taiwan, 2014, pp. 1-4.

[30] Shiwei Luan, D. Gude, P. Prakash, and S. Warren, "A paraeducator glove for

counting disabled-child behaviors that incorporates a Bluetooth Low Energy wireless link to a smart phone," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Medicine Biology Soc.*, Chicago, IL, USA, 2014, pp. 796-799.

[31] E. Mackensen, M. Lai, and T.M. Wendt, "Bluetooth Low Energy (BLE) based wireless sensors," in *Proc. IEEE SENSORS*, Taipei, Taiwan, 2012, pp. 1-4.

[32] B.K. Maharjan, U. Witkowski, and R. Zandian, "Tree Network Based on Bluetooth 4.0 for Wireless Sensor Network Applications," in *Proc. 6th European Embedded Design Education Research Conf.*, 2014.

[33] J. Masuch and M Delgado-Restituto, A 190-microWatt Zero-IF GFSK Demodulator with a 4-b Phase-Domain ADC, *IEEE J. Solid-St. Circ.* **47** (2012), 2796-2806.

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

[35] J. Nieminen et al., Networking Solutions for Connecting Bluetooth Low Energy Enabled Machines to the Internet of Things, *IEEE Network* **28** (2014), 83-90.

[36] J. Nieminen, T. Savolainen, M. Isomaki, Z. Shelby, and C. Gomez, "Transmission of IPv6 packets over Bluetooth Low Energy," draft-ietf-6lowpan-btle-12, 2013.

[37] M. Park, "Proposed Specification Framework for TGah," 2012.

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

[39] Qiao Qiao, Fei Gao, and Te Wang, "The design of electronic thermometer based on Bluetooth Low Energy," in *Proc. 4th Int. Conf. Electron. Inf. Emergency Commun.*, Beijing, China, 2013, pp. 252-254.

[40] K. Mikhaylov, Accelerated Connection Establishment (ACE) Mechanism for Bluetooth Low Energy, in: Proc. IEEE 25th Annual Int. Symp. Personal Indoor Mobile Radio Commun., Washington, DC, USA, pp. 1264-1268.

[41] K. Mikhaylov, Accelerated Connection Establishment (ACE) Mechanism for Bluetooth Low Energy, in: Proc. IEEE 25th Annual Int. Symp. Personal Indoor Mobile Radio Commun., Washington, DC, USA, pp. 1264-1268.

[42] K. Mikhaylov, "Simulation of Network-Level Performance for Bluetooth Low Energy", in Proc. IEEE 25th Annual Int. Symp. Personal Indoor Mobile Radio Commun., Washington, DC, USA, pp. 1259-1263

[43] K. Mikhaylov, N. Plevritakis and J. Tervonen, Performance analysis and comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI, J. Sens. Actuator Networks, 2(2013), 589–613.

[44] K. Mikhaylov and J. Tervonen, Analysis and evaluation of the maximum throughput for data streaming over IEEE 802.15.4 wireless networks, J. High Speed Networks 19(2013), 181-202.

[45] O. Galinina, K. Mikhaylov, S. Andreev, A. Turlikov and Y.Koucheryavy, Smart Home Gateway System over Bluetooth Low Energy with Wireless Energy Transfer Capability, EURASIP J. Wireless Commun. Networking, 2015:178.

[46] H.J. Rivertz, "Bluetooth Security," DART/05/05 2005.

[47] M. Ryan, "Bluetooth: With Low Energy Comes Low Security," in *Proc. 7th USENIX Workshop Offensive Techn.*, Washington, D.C., USA, 2013, pp. 1-7.

[48] M. Siekkinen, M. Hiienkari, J.K. Nurminen, and J. Nieminen, "How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee / 802.15.4," in *Proc. IEEE Wireless Commun. and Networking Conf. Workshops*, Paris, France, 2012, pp. 232-237.

[49] H. Strey, P. Richman, R. Rozensky, S. Smith, and L. Endee, "Bluetooth Low Energy Technologies for Applications in Health Care: Proximity and Physiological Signals Monitors," in *Proc. 10th Int. Conf. Expo Emerging Technologies Smarter World*, Melville, NY, USA, 2013, pp. 1-4.

[50] N. Ullah, M.S. Chowdhury, P. Khan, and K.S. Kwak, "Throughput limits of UWB based 802.15.4a," in *Int. Conf. Inf. Commun. Tech. Convergence* , Jeju, ROK, 2010, pp. 166-167.

[51] H. Wang, M. Xi, J. Liu, and Chen C., "Transmitting IPv6 packets over Bluetooth low energy based on BlueZ," in *Proc. 15th Int. Conf. Adv. Commun. Technol.*, Pyeongchang, ROK, 2013, pp. 72-77.

[52] A. Wong et al., "A 1 V 5 mA Multimode IEEE 802.15.6 / Bluetooth Low-Energy WBAN Transceiver," in *Proc. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, USA, 2012, pp. 300-302.

[53] B. Yu, L. Xu, and Y. Li, "Bluetooth Low Energy (BLE) Based Mobile Electrocardiogram Monitoring System," in *Proc. Int. Conf. Inf. and Automation*, Shenyang, China, 2012, pp. 763-767.

[54] Y. Zhang, A. Atac, L. Liao, and S. Heinen, "A Low-Power High-Efficiency Demodulator in Bluetooth Low Energy Receiver," in *Proc. 8th Conf. Ph.D. Research Microelectronics Electronics*, Aachen, Germany, 2012, pp. 1-4.

[55] Menghan Zhang, Weiwei Xia, and Lianfeng Shen, "Bluetooth Low Energy Based Motion Sensing System," in *Proc. 6th Int. Conf. Wireless Commun. Signal Processing*, Hefei, China, 2014, pp. 1-5.

[56] Yang Zhao and J.B. Laflen, "Gesture Recognition with Low Quality Signal over Low Energy Bluetooth 4.0," in *Proc. IEEE Int. Conf. Consumer Electron.*, Las Vegas, NV, USA, 2014, pp. 294-295.

[57] Bing Zhou et al., "A Bluetooth low energy approach for monitoring electrocardiography and respiration," in *Proc. IEEE 15th Int. Conf. Health Networking, Applications & Services*, Lisbon, Portugal, 2013, pp. 130-134.

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