TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 3486 – 3499
© TÜBİTAK
doi:10.3906/elk-1901-109

Increasing Bluetooth Low Energy communication efficiency by presetting protocol parameters

Dušan HATVANI, Dominik MACKO*
Institute of Computer Engineering and Applied Informatics, Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava, Bratislava, Slovakia

Received: 16.01.2019 • **Accepted/Published Online:** 24.05.2019 • **Final Version:** 18.09.2019

Abstract: Standard protocols are important regarding the compatibility of devices provided by different vendors. However, specific applications have various requirements and do not always need all features offered by standard protocols, making them inefficient. This paper focuses on standard Bluetooth Low Energy modifications, reducing control overhead for the intended healthcare application. Specifically, the connection establishment, device pairing, and connection parameter negotiations have been targeted. The simulation-based experiments showed over 20 times reduction of control-overhead time preceding a data transmission. It does not just directly increase the energy efficiency of communication; it also prolongs the time for sensor-based end devices to spend in an energy-saving mode. The result is a longer runtime of such sensor devices powered by batteries.

Key words: Bluetooth Low Energy, energy efficiency, Internet of Things, low power, wireless sensor networks

1. Introduction

Emerging fields of the Internet of Things (IoT) put specific requirements on the standardly used communication technologies and architectures. Connecting the previously unconnected network types, such as body area networks or wireless sensor networks, to the Internet creates new challenges. Energy-constrained (e.g., battery-operated or energy-harvesting) devices must now support new features to be connected to and managed from other Internet-connected devices. This makes the energy efficiency of such devices and supported communication technologies a cornerstone.

Energy-constrained sensor end devices are usually focused on a specific task of their application and are usually connected to some specific gateway or relay to connect them to the rest of the network (e.g., the Internet). Therefore, these devices usually always use the same communication technology and always communicate with the same next-hop device with always the same connection parameters. Therefore, they cannot utilize all the features offered by standard protocols, such as Bluetooth Low Energy (BLE) [1].

Many existing works deal with energy efficiency in the wireless sensor networks or IoT area. Based on the survey in [2], the existing techniques can be classified as data reduction (e.g., [3, 4]), protocol overhead reduction (e.g., [5]), energy-efficient routing (e.g., [6]), duty cycling (e.g., [7, 8]), and topology control (e.g., [9, 10]). Even the newest incarnation of Bluetooth, specifically version 5 [11], targets IoT devices. It offers a larger range and higher speed than previous versions while reducing energy requirements [12]. However, there are scenarios in which the older version (BLE 4.2) is still more energy-efficient. Moreover, BLE 5 requires new

*Correspondence: dominik.macko@stuba.sk

chips to be integrated into the existing devices in order to utilize the benefits. Since it takes time for BLE 5 to become massively deployed, there is still space for improvements of BLE 4.2.

This work is focused on reducing control overhead of the standard BLE. Key contributions of this work in this sphere are summarized below.

- Modified connection establishment – by scanning all three advertising channels simultaneously, the probability of a channel-match between communicating devices is increased and the connection-establishment time is reduced.
- Optimized device pairing and selection of an association model – by reducing negotiations during pairing and association phases, the energy required for sending and processing of multiple messages is spared and the duration time of these phases is reduced.
- Preset connection parameters and data length extension (DLE) – by always accepting connection parameters offered by the master device (they are calculated based on the application), further negotiations are eliminated.

The primary idea is to reduce negotiations for applications, in which the parameters can be predetermined. The reduced negotiations would mean that fewer messages need to be exchanged between devices, which reduces the energy required for processing these messages. Moreover, less control overhead also reduces the time for which the energy-constrained sensor devices need to be active. Spending more time in energy-saving mode prolongs the battery life. Reduction of negotiations also eliminates the communication noise for other communications.

The rest of the paper is organized as follows. In Section 2, individual contributions are described in more detail. Section 3 is focused on discussion about the obtained results. Finally, the conclusion is given in Section 4.

2. The proposed methods for improving BLE efficiency

The intended application targeted in this work belongs to a healthcare-monitoring area of the Internet of Things. It assumes that sensor end devices will be located on patients' bodies and will be powered by batteries. They will be used for monitoring life functions, such as heart rate, blood pressure, stress, breath, and temperature. The data will be harvested periodically and also event-triggered (e.g., heart or breath stop, fall, rapid increase in stress or temperature). The sensor data will be collected by central devices using wireless communication, which will be powered by an electrical grid. The goal is to increase the time that sensor end devices will run on batteries. The result would reduce the amount of manually changing their batteries, limit downtime (time during which a patient is not sensor-monitored), and thus increase the comfort of patients.

2.1. Connection establishment

Bluetooth Low Energy [13] uses 37 channels for data traffic (channel numbers of 0 to 36) and three channels for control traffic (channel numbers of 37, 38, and 39). The control traffic includes device discovery, connection establishment, and broadcasts (the control-traffic channels are often referred to as advertising channels). A device can send messages via an advertising channel in time intervals called advertising events. During such an event, the device sequentially uses all three advertising channels to transmit the message (in order to increase the probability of avoiding interference). The receiving device must scan the corresponding advertising channel in order to receive the message (e.g., for connection establishment). This means that there must be a match between advertising and scanning channels for some period of time to successfully transmit the message. Therefore, some power is wasted to unnecessarily transmit the message via redundant channels.

The first proposed modification lies in the idea that a scanning device will have three BLE modules, each of them scanning a different dedicated advertising channel. In the intended application, a scanning device is represented by a central node, which will be powered by a grid, and thus increased power requirements on the scanning side is not a problem. The scanning interval and scanning window should be according to the BLE standard less than 10.24 s due to channel alternation. The proposed modification cancels these values, since all three channels are scanned continually. Thus, an advertising device (an energy-constrained sensor end node) will randomly select a single advertising channel to transmit the ADV_IND message during an advertising event. Since the proposed modification does not alternate advertising channel for the duration of such an event, the advertising event can be assigned a 30 ms time period in this channel, instead of the standard period of up to 10 ms for each advertising channel. If the selected channel is congested (e.g., communication noise) and the advertising device does not get a response, a simple timer can change the advertising channel. If all three channels are congested (i.e. the devices unsuccessfully tried all three advertising channels), the device is delayed for a period of three times the duration of an advertising event to reduce wasted energy.

If congestion is not taken into account, the proposed modification should enable establishing a connection immediately. However, to verify such a hypothesis, there is a need to express this process analytically. To establish a connection between a single scanning device and a single advertising device, two conditions have to be met. The first one is a communication channel match and the second condition is an in-time exchange of connection-establishment messages. Communication noise is not included in the calculations. According to [14], the probability P of meeting these two conditions can be expressed by

$$P = \frac{1}{3} \times \left[\rho - \left(\frac{T_S}{\tau_{SI}} \right) \right], \tag{1}$$

where $\frac{1}{3}$ refers to the probability of a match between communication channels of the advertising and scanning nodes, and $\rho - \left(\frac{T_S}{\tau_{SI}} \right)$ represents a probability of successful exchange of messages to establish a connection. The ρ parameter is a ratio of $\frac{\tau_{SW}}{\tau_{SI}}$, where τ_{SW} refers to an advertising period and τ_{SI} refers to a scanning period. This parameter is called a duty cycle, which expresses how much time a device spends on scanning. The T_S parameter represents the time period required to successfully establish a connection.

In the proposed method, a single channel is used for advertising; however, all three channels are scanned simultaneously. This means that the first condition is always met. Since there are no advertising and scanning periods in the proposal, the exchange of connection-establishment messages is always on time (i.e. successful). To summarize, using the proposed method, the probability $P = 1$, i.e. conditions are met by 100% (when noise or collisions are not counted) and the time required for connection establishment is equal to the time required for exchanging control messages (T_S).

2.2. Pairing and association

Pairing is a process of exchanging the security key and authentication. The BLE pairing is divided into three phases. In the first phase, the devices exchange parameters for pairing. These parameters specify what association model should be used [15]. They mainly include input/output capabilities (i.e. if a device has a keyboard or buttons and some sort of display) and expected security features (e.g., to prevent a man-in-the-middle attack). There are basically four association models in BLE: numeric comparison (i.e. a user must confirm that numbers displayed by both devices are equal), passkey entry (i.e. a user must enter into one device the passkey generated by the other device or a user must enter an equal passkey into both devices), out of band (i.e. a portion of the security key is transmitted via another communication technology), and just works (i.e. without security features, for devices without input/output capabilities).

The intended application domain requires a secure connection. Therefore, the just works association model cannot be used. Another requirement is that the manual comparison during each new connection between devices is avoided. Moreover, energy-constrained sensor devices cannot include a different technology interface for an out-of-band communication. Thus, no existing association model suits these needs. The second proposed BLE modification uses a modified numeric-comparison association model. A sensor device can have a small display to show the calculated numerical value (a six-digit display is enough) and a button to acknowledge the equality of numbers at both devices. In case the received number is not acknowledged, the connection attempt is repeated. In order to avoid manual checking upon a failure of an existing connection (e.g., temporarily out of range), both devices can save the required parameters into the memory. Such an association model is known beforehand, as well as the values of other important pairing parameters. Specifically for the intended application, input/output capabilities of the used devices are always the same, the SC flag for a secure BLE connection is always set to 1, and the man-in-the-middle protection is always activated. Since all of these important parameters can be predefined, negotiations in the first pairing phase are redundant and can be skipped directly to the second phase.

The time required for exchanging messages during the first pairing phase T_P is equal to a sum of time for sending a request message T_{REQ}, a time for sending a response message T_{RSP}, and a time period between receiving a message and sending another message T_{IFS}:

$$T_P = T_{REQ} + T_{IFS} + T_{RSP}. \tag{2}$$

The modified pairing method uses predefined pairing parameters. This means that during the first pairing phase, there are no messages exchanged, which spares the entire time T_P as well as the energy required for processing and sending of the REQ and RSP messages.

The second pairing phase serves for exchange of corresponding public keys, for establishment of Diffie–Hellman keys, and for exchange of confirmation and corresponding random numbers to calculate the six-digit numbers by each side to compare. If a user manually acknowledges the equality, a long-term key for encryption is generated, which is used in the third phase to secure the channel. Since the encryption starts in the third phase, the authentication has a weakness when an attacker can reveal the association model and its possible vulnerability.

For the second phase, there is a possibility to modify the authentication process and reduce the manual intervention completely. In such a case, the second phase would end after checking the confirmation value, and then the generation of the long-term key for the third phase would proceed. Both authentication options have pros and cons. The first option offers stronger security due to manual checking of calculated numbers; however, the authentication is longer by several steps. The amount of exchanged messages is unchanged in comparison to standard BLE and thus the time requirements are approximately the same. The second option (a lightweight model) spares energy due to shorter authentication and reduces cost of end devices due to not using displays and buttons (which reduces energy requirements even more). The time spared using this approach is subjective, since the checking is manual. The downside is a lower security, which is however compensated by no communication in the first phase, which makes it harder for an attacker to reveal the association model. Since this work is focused on reducing energy requirements of end devices as much as possible, the second option is preferred for the intended application.

2.3. Connection parameters

There are several parameters that influence the BLE connection between the devices. A connection interval, a slave latency, and a connection supervision timeout are among the most important. The central node, acting as a master for this connection, decides how often the synchronization occurs. The end node, acting as a slave, cannot decide about the connection parameter; however, it can suggest some values. The central node selects the connection-interval value (either taking into account the end node's suggestions or not), which represents a time when devices can send and receive data. The slave latency implies how many times the end device can ignore a connection event (in order to save the power). Upon the next connection event the end device must respond; otherwise, the connection can be lost. The decision of whether the connection is lost is based on the supervision timeout (how much time the device may not hear the peer before canceling the connection).

The selection of the value for the connection interval should imply the most efficient data exchange. It means that all data (needed by the intended application) must be transferred while reducing the idle time within such an interval. The idle time is mostly reduced when the time interval corresponds to the data-exchange time. The number of required RX and TX exchanges can be deduced based on the data length, which is predictable in the intended application. At most, 251 B of data can be transmitted in a single TX message. The value of the connection interval, however, relates to other parameters. In the intended application, the data can be sent when ready in order to save the most energy of the sensor end device. Thus, the slave latency is the highest possible number. The supervision timeout ST is then determined as a connection interval T_{CI} multiplied by slave latency SL increased by 1:

$$ST = T_{CI} \times (SL + 1). \tag{3}$$

Based on these presumptions, the third proposed modification expects the sensor end device not to suggest any parameters. They are calculated by the central device based on the expected data length. A single message is then enough to send the final connection parameters by the central device to the end device.

Similarly to the first pairing phase, a negotiation of the DLE includes an exchange of two messages, namely LL_LENGTH_REQ and LL_LENGTH_RSP. The time T_{DLE} required for such an exchange can be expressed by:

$$T_{DLE} = T_{LL\_LENGTH\_REQ} + T_{IFS} + T_{LL\_LENGTH\_RSP}. \tag{4}$$

Since the payload length is predefined in the proposed method, the exchange of these messages is not occurring, and thus the corresponding time and energy are spared.

A very similar result is achieved for a negotiation of connection parameters. Although some communication is still required for this process, there is time and energy required for the processing and sending of two messages spared. The time T_{CP} required for the negotiation of connection parameters can be expressed by:

$$T_{CP} = T_{ICP} + T_{IFS} + T_{SREC} + T_{IFS} + T_{FCP}, \tag{5}$$

where T_{ICP} represents the time required for sending the initial parameters by the central node, T_{SREC} refers to the time required for sending a suggestion of recommended parameters by the sensor end node, and T_{FCP} is the time required for sending the final parameters by the central node.

The proposed method expects only the final parameters to be sent, which means that the time required for the negotiation of connection parameters is equal to the time for sending the final parameters:

$$T_{CP} = T_{FCP}. \tag{6}$$

The proposed method can predict the connection interval based on the length of the received data. Let us assume that the time length of the connection interval is given by:

$$T_{CI} = n \times (T_{RX} + T_{IFS} + T_{TX} + T_{IFS}) + T_D. \tag{7}$$

This equation implies that the length of the connection interval is equal to a sum of a time required for n exchanges of RX and TX messages $(n \in N, n \geq 1)$ and a time TD, during which the messages cannot be exchanged. It means that $0 \leq T_D < (T_{RX} + T_{TX} + 2 \times T_{IFS})$. Since the connection interval is predicted based on the data length, the time T_D will be 0 or it will be approaching this value.

Reduction of the number of exchanged messages does not just spare the time and energy of the communicating devices, but also eliminates the communication noise for other communications.

3. Results and discussion

The proposed modifications of the standard BLE protocol and processes have been validated and evaluated using two approaches (in addition to the analytical evaluation outlined in the previous section): Petri nets and dedicated BLE simulations.

For Petri net modeling and simulations, a tool called PIPE v4.3.0 [16] has been used. This simulator offers modeling and testing of colored Petri nets with optional timed transitions. The colored tokens help to differentiate the actions required for a transition to another place. The blue tokens in the models represent a state, the red tokens represent an exchange of messages, and the black tokens represent other common actions.

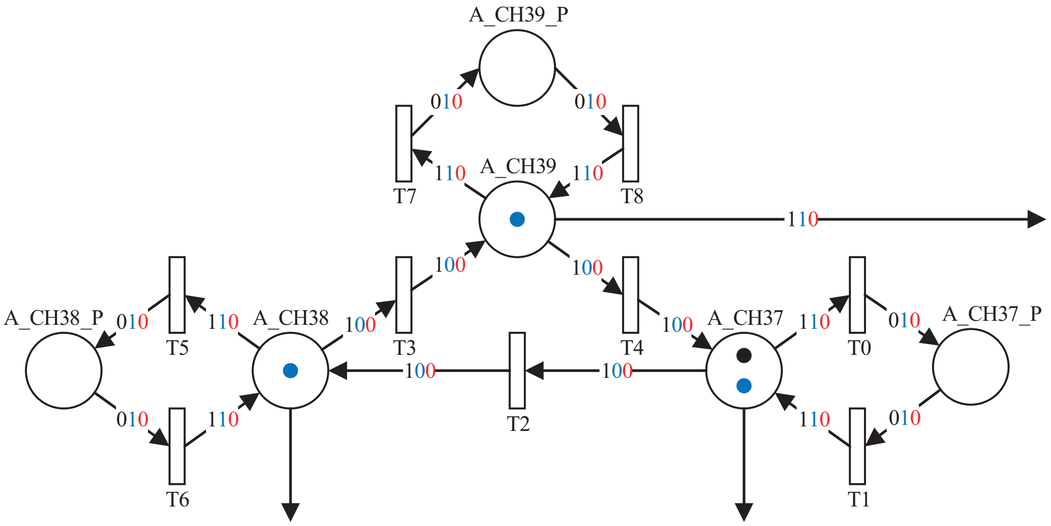


Figure 1. Advertising part of the Petri net model for the standard BLE.

Figure 1 illustrates the advertising part of the created Petri net model for the standard BLE. The scanning part is very similar to this one. The places in the figure represent the active and passive (the place name ending with "_P") states of individual channels. The black tokens serve for illustration of the currently used channel and the blue tokens indicate the state of each channel. A channel always starts in the active state and its state can change only if the channel is currently used. The same rules applies to the scanning part. For connection establishment, channels at both parts must match (i.e. advertising and scanning devices use the same channel) and also must be in the active states. After meeting these conditions (i.e. the connection is established), the communication continues with the device-pairing phase. This phase represents an exchange of control messages containing the pairing parameters between the advertising and the scanning node. The successful exchange of pairing messages is followed by the authentication phase, which uses the previously exchanged parameters. The model works analogously during negotiations of connection parameters and DLE. Simulation ends by successful selection of DLE, which means that the data transfer can begin. This process is illustrated in Figure 2. Individual phases require exchange of multiple messages, the number of which is indicated by the red-colored numbers in the model (for arrows outgoing from exchange/transfer places). The model is designed in a secured way that prevents triggering of transitions for the finished phases.

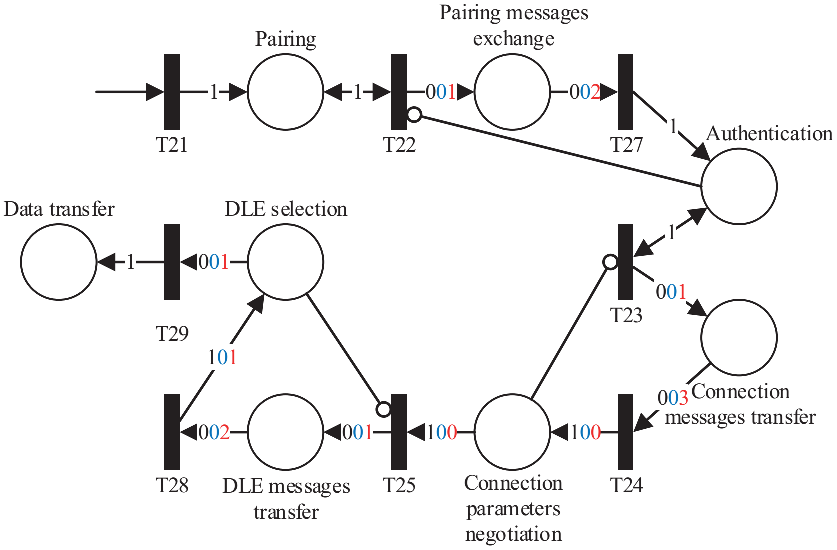


Figure 2. Pairing and parameters selection part of the Petri net model for the standard BLE.

The simulation results of such a model reported by the used tool for 1000 transition firings and 100 replications are provided in Table 1. The constructed coverability graph is illustrated in Figure 3.

The lightweight model of the modified BLE according to the proposed methods is illustrated in Figure 4. The scanning phase is simplified due to scanning of all three channels simultaneously. Also, the exchange of several parameters was eliminated, or at least the number of exchanged messages was reduced. Because of that, the model does not require counting of exchanged messages or securing the transitions in individual phases.

The simulation results of the lightweight model reported by the PIPE tool for 1000 transition firings and 100 replications are provided in Table 2. The corresponding constructed coverability graph is illustrated in Figure 5.

Although the used Petri net simulator offers utilization of colored Petri nets and timed transitions, it is not suitable for simulation of a protocol that requires real-time transitions. The problem is that it is not possible to simulate a selected transition, because the transitions are triggered randomly. This means that if multiple transitions can be triggered in a given time, only a random one from them is actually triggered. Nevertheless, simulations in this tool are sufficient to verify a connection establishment process as well as the exchanges of

Table 1. Standard BLE petri net simulation results.

Place	Average number of tokens	95% confidence interval (+/-)
A_CH37	0.001	0
A_CH37_P	0	0
A_CH38	0	0
A_CH38_P	0	0
A_CH39	0	0
A_CH39_P	0	0
Authentication	0.999	0
Connection establishment	0	0
Connection messages transfer	0	0
Connection parameters negotiation	0.999	0
Data transfer	15.98402	0
DLE messages transfer	0	0
DLE selection	16.98302	0
Pairing	0.999	0
Pairing messages exchange	0	0
S_CH37	0.001	0
S_CH37_P	0	0
S_CH38	0.84515	0.02344
S_CH38_P	0.15485	0.02344
S_CH39	0.86513	0.02284
S_CH39_P	0.13487	0.02284

Table 2. Standard BLE petri net simulation results.

Place	Average number of tokens	95% confidence interval (+/-)
Advertising	0.01408	0
Authentication	0	0
Connection establishment	0	0.03541
Connection parameters negotiation	33.98592	0.13136
Data transfer	34.02817	0.07984
Scanning	0.01408	0

control messages. The tool was used to prove that the data transfer state can be successfully reached. It was also used to show a comparison between complexity of the standard and modified BLE protocols.

For verification of time requirements and efficiency of the proposed methods in comparison to the standard BLE communication, a new time simulator has been developed. This simulator is able to evaluate the time required for connection establishment as well as for subsequent exchange of control messages. The simulator has been implemented as a computer application using scripts in the Python 3 programming language. One script simulates the advertising device and another script simulates the scanning device. Two pieces of the

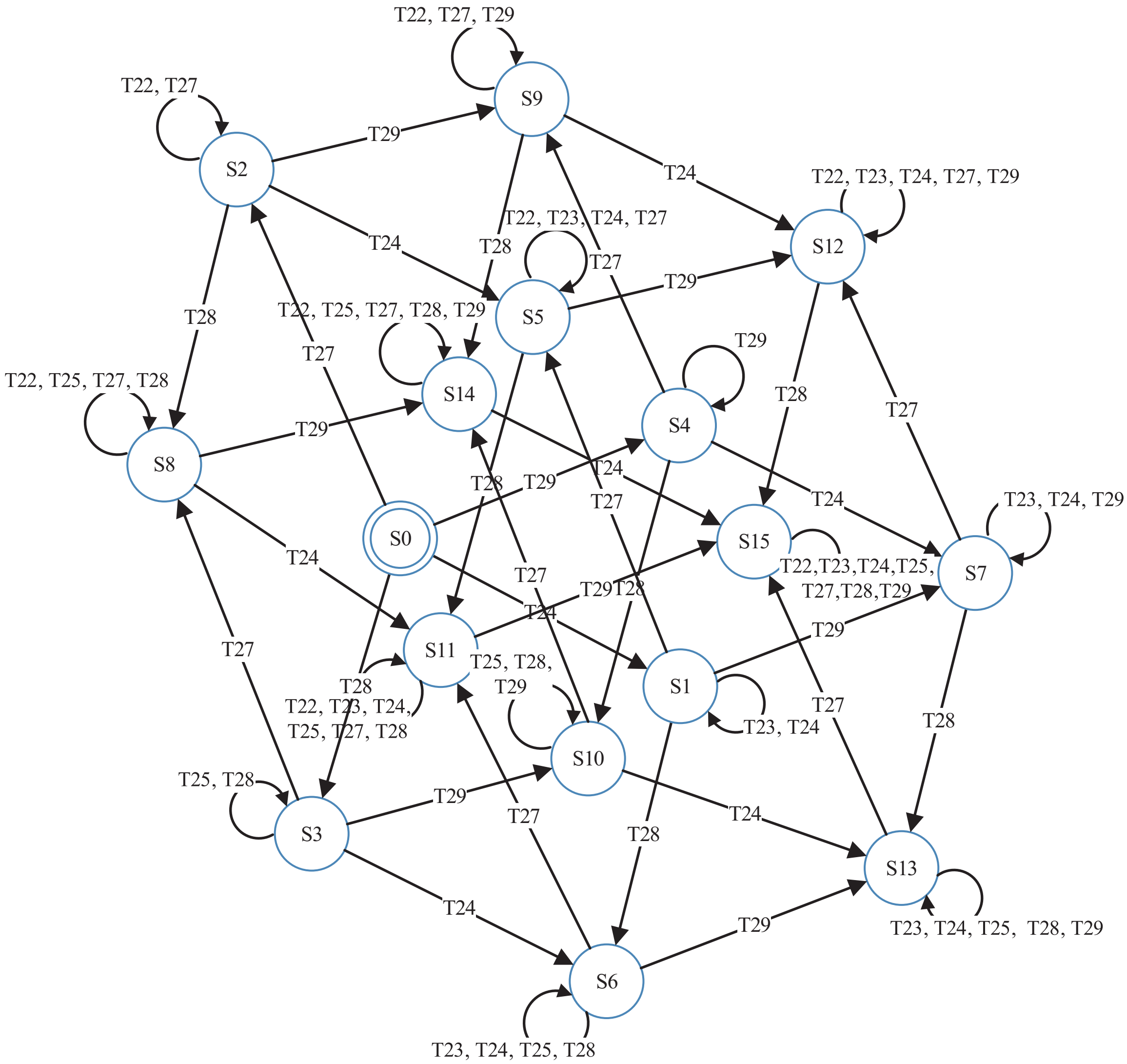


Figure 3. Reachability/coverability graph of the standard BLE Petri net model.

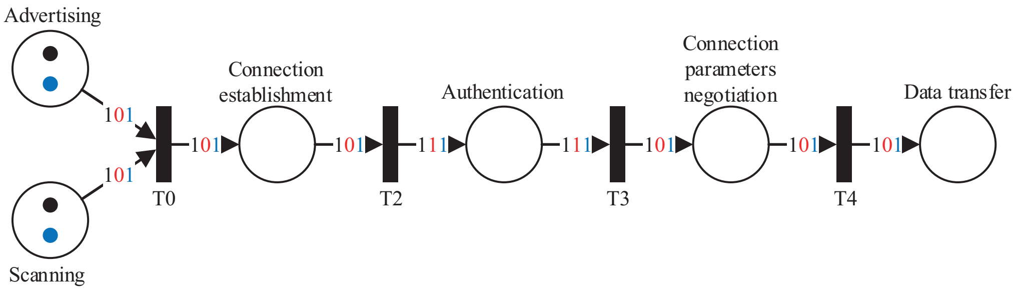


Figure 4. Petri net model for the modified BLE using the proposed methods.

advertising-device script are provided in Figure 6: the advertising channel change algorithm is contained in Figure 6a and the advertising delay and interval simulation algorithm is illustrated in Figure 6b. The scanning device is changing the scanning channel analogously.

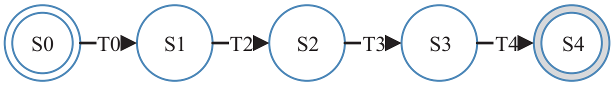
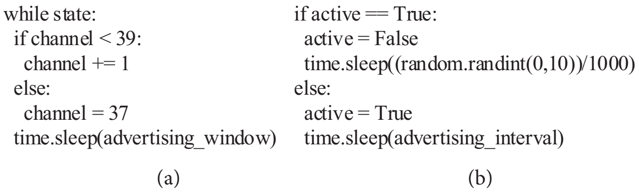


Figure 5. Reachability/coverability graph of the modified BLE Petri net model.



```
while state:
 if channel < 39:
  channel += 1
 else:
  channel = 37
 time.sleep(advertising_window)
```

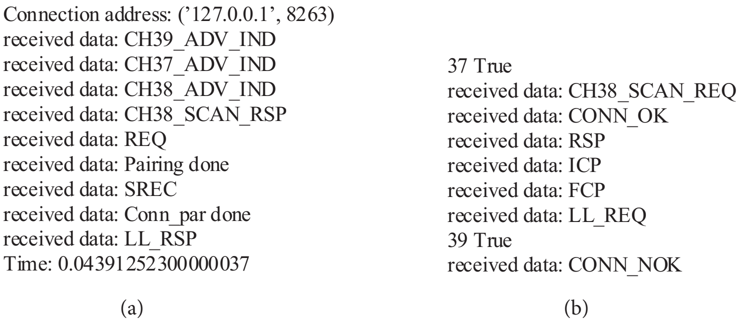
(a)

```
if active == True:
 active = False
 time.sleep((random.randint(0,10))/1000)
else:
 active = True
 time.sleep(advertising_interval)
```

(b)

Figure 6. (a) Advertising channel change algorithm. (b) Advertising delay and interval simulation algorithm.

In the simulations, these devices actually exchange BLE messages via a TCP connection, which ensures correct ordering of the messages. In order for such unrealistic overhead to be excluded from the resulting time requirements, the time measurements start after the creation of the TCP connection and the measurements end after the successful receiving of the last BLE control message. A log file is used to review the messages that were exchanged during the simulation, along with the time required for exchange of these BLE messages. An example of a scanning device log file is provided in Figure 7a and an advertising device log file is illustrated in Figure 7b.



```
Connection address: ('127.0.0.1', 8263)
received data: CH39_ADV_IND
received data: CH37_ADV_IND
received data: CH38_ADV_IND
received data: CH38_SCAN_RSP
received data: REQ
received data: Pairing done
received data: SREC
received data: Conn_par done
received data: LL_RSP
Time: 0.04391252300000037
```

(a)

```
37 True
received data: CH38_SCAN_REQ
received data: CONN_OK
received data: RSP
received data: ICP
received data: FCP
received data: LL_REQ
39 True
received data: CONN_NOK
```

(b)

Figure 7. (a) Scanning device log file example. (b) Advertising device log file example.

There are several BLE parameters that must be preset before a simulation, namely the advertising window, the advertising interval, and the scanning window (the scanning interval is the same as the scanning window). The advertising delay is a pseudorandom value up to 10 ms, as predefined by the standard. The standard also defines the ranges of usable values for the configurable parameters. The advertising interval can be a value in the range from 20 ms to 10.24 s, in steps of 0.625 ms. The advertising window is a value from 0 to 10 ms. Finally, the scanning window can be a value in the range from 2.5 ms to 10.24 s (also in steps of 0.625 ms).

First, these parameters were fixed. The advertising window of 10 ms has been used, the advertising interval has been set to 325 ms, and the scanning window of 12.5 ms has been used. The only variable parameter was selection of the initial channels at both simulated devices. The tests included all possible combinations of initial channels at the advertising and the scanning device. For each combination, multiple simulations were performed (i.e. five to ten runs to eliminate the effect of the system processing other tasks) and the average time requirement was calculated. Since the simulation time is dependent on the device that runs the simulation, the tests were performed using three computer devices (i.e. stations[1]) with different performances in order to objectively compare the results. The results of the simulated standard BLE are reported in Table 3.

Table 3. Average results of the simulated standard BLE (time in units of ms).

Station	37/37	37/38	37/39	38/37	38/38	38/39	39/37	39/38	39/39
1	10.810	52.930	101.378	101.855	10.807	55.741	52.809	96.987	10.407
2	23.034	372.237	588.953	316.811	42.156	448.948	520.616	665.367	20.806
3	41.260	561.729	338.468	324.509	37.508	371.908	243.835	466.635	37.607

The first column contains the station identifier and the other columns represent combinations of initial scanning/advertising channels and the corresponding average time (in units of ms). The results of the simulated modified BLE are reported in Table 4, along with a comparison to the standard BLE. The second column represents the average time of the proposed BLE modification for each station. The third column represents the minimal value reported in Table 3 divided by the value of the proposal for each station. The last column presents the spared time as a difference between the minimal value reported in Table 3 and the value of the proposal for each station.

Table 4. Average results of the simulated modified BLE and its comparison to the simulated standard BLE.

Station	Proposal [ms]	$\frac{MIN(Table\ 3)}{Proposal}$	Spared time [ms]
1	0.529	19.674	9.878
2	0.735	28.315	20.071
3	1.300	28.855	36.208

As one can notice, the proposed solution is more efficient. Compared to an ideal situation in the standard BLE, when the advertising device selects the same initial channel as the scanning device, the proposed modification requires at least 20 times less time to exchange the control messages needed before data transfer can start. To better illustrate the difference, the results are shown in the chart provided in Figure 8 (the vertical axis is provided in a logarithmic scale).

In order to evaluate the effect of these results on the energy consumption, we can use a rough estimation by an example device power profile. According to [17], the SPW-2 device has an idle power consumption of 8.6 μW and approximately 20 mW when advertising at 5 dBm. The spared energy can be computed as a multiplication of a difference between advertising power and idle power (=19.9914 mW) and the spared time during connection initiation. For the previously mentioned ideal situation in the standard BLE (i.e. the channel match), the values of the spared time from Table 4 can be used. The spared energy is then 197.5 μJ for Station1, 401.2 μJ for Station2, and 723.9 μJ for Station3 during each connection initiation process. The real spared energy would be even higher.

To show that the benefits of the proposed modification are not tightly coupled with the used simulation parameters, we conducted another experiment in which we used different parameters. The simulation was performed using Station3 only. The parameters were scaled from the minimum allowed by the standard BLE

[1] Station1: Intel Core i7, Ubuntu 16 64-bit; Station2: Intel Core i3, Windows 10 Home 64-bit; Station3: Intel Core2, Windows 7 Enterprise 32-bit.

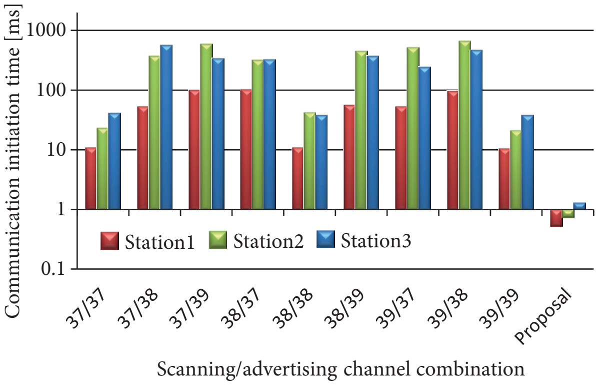


Figure 8. Comparison of results obtained from the implemented BLE simulator.

to the maximum. In addition, two sets of parameters in between those boundaries were used. The results are provided in Table 5. The first column contains a set of parameters (AI represents the advertising interval, AW represents the advertising window, and SW represents the scanning window), the next columns represent individual combinations of initial scanning/advertising channels (similarly to Table 3), and the last column represents the minimal value from each row divided by the value of the proposal for Station3 (1.3 ms, as reported in Table 4). The results of this experiment show that the benefits of the proposed BLE modification are obvious also when other values of the configurable simulation parameters are used.

Table 5. Results comparison of various simulation parameters using Station3.

Parameters	37/37	37/38	37/39	38/37	38/38	38/39	39/37	39/38	39/39	$\frac{MIN}{Proposal}$
AI=20 ms, AW=0 ms, SW=2.5 ms	86.332	84.090	62.673	53.744	44.270	44.312	47.443	46.610	55.560	34.056
AI=325 ms, AW=10 ms, SW=12.5 ms	41.260	561.729	338.468	324.509	37.508	371.908	243.835	466.635	37.607	28.855
AI=1 s, AW=5 ms, SW=5 s	57.432	46.155	63.115	62.921	60.723	76.564	52.187	91.693	60.276	35.507
AI=10.24 s, AW=10 ms, SW=10.24 s	53.997	39.054	64.491	64.428	49.081	56.057	35.868	64.464	64.709	27.593

These are, however, small units of time, which are unnoticeable for a human being. Nevertheless, the communication using the proposed solution is more efficient in terms of time, as well as the energy (since there is a smaller amount of bits required to transmit). It can prolong the device lifetime running on a battery. Since the results are obtained using the simulation only, the values can be influenced by side-effects (e.g., thread handling in the simulator) and thus not completely accurate. However, processing of messages on a real device would also be influenced by such side effects. Further testing should use over-the-air BLE communication. The most difficult to achieve is the first proposed modification, since it would require a modification of the BLE chip hardware in order to enable scanning of all three channels at the same time. This is too expensive for us at this time. The other proposed modifications requires tampering with the BLE firmware (driver/library) only, which is easier to achieve. Therefore, such experiments represent the next step in this research.

4. Conclusion

In this paper, three modifications to the standard Bluetooth Low Energy have been proposed. The work was targeted towards control communication. Based on the presetting of some important parameters and reduction of unnecessary negotiations, the energy efficiency of the BLE control communication has been increased for the intended healthcare application. The simulation-based experiments showed over 20 times reduction of time required for the control overhead preceding a data transmission, i.e. reducing the wasted energy. Moreover, a shorter communication time means that the energy-constrained sensor end devices can spend more time in the energy-saving mode, prolonging their battery-operation lifetime even more.

Acknowledgments

This work was supported by the Ministry of Education, Science, Research, and Sport of the Slovak Republic within the Research and Development Operational Programme for the project ”University Science Park of STU Bratislava”, ITMS 26240220084, co-funded by the European Regional Development Fund. The work was also supported by the Slovak Research and Development Agency (APVV-15-0789), the Slovak Cultural and Educational Grant Agency (KEGA 011STU-4/2017), and projects no. 2018/14427:1-26C0 and 002STU-2-1/2018. D.H. did the experiments, D.M. supervised the work, and both authors wrote the paper.

References

[1] Bluetooth Special Interest Group. Specification of the Bluetooth System. Covered Core Package version: 4.2. Bluetooth SIG, 2014.

[2] Soua R, Minet P. A survey on energy efficient techniques in wireless sensor networks. In: 2011 4th Joint IFIP Wireless and Mobile Networking Conference; Toulouse, France; 2011. pp. 1-9.

[3] Alippi C, Anastasi G, Di Francesco M, Roveri M. Energy management in wireless sensor networks with energy-hungry sensors. IEEE Instrumentation & Measurement Magazine 2009; 12 (2): 16-23. doi: 10.1109/MIM.2009.4811133

[4] Kolios P, Ellinas G, Panayiotou C, Polycarpou M. Energy efficient event-based networking for the Internet of Things. In: 2016 IEEE 3rd World Forum on Internet of Things; Reston, VA, USA; 2016. pp. 1-6.

[5] Feng H, Ma L, Leng S. A low overhead wireless sensor networks MAC protocol. In: 2010 2nd International Conference on Computer Engineering and Technology; Chengdu, China; 2010. pp. V4-128-V4-131.

[6] Ogundile OO, Alfa AS. A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks. Sensors 2017; 17 (5): 1084. doi: 10.3390/s17051084

[7] Fafoutis X, Tsimbalo E, Piechocki R. Timing channels in Bluetooth Low Energy. IEEE Communications Letters 2016; 20 (8): 1587-1590. doi: 10.1109/LCOMM.2016.2574311

[8] Carrano RC, Passos D, Magalhaes LCS, Albuquerque CVN. Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. IEEE Communications Surveys Tutorials 2014; 16 (1): 181-194. doi: 10.1109/SURV.2013.052213.00116

[9] Chmaj G, Selvaraj H. Energy-efficient distributed computing solutions for Internet of Things with ZigBee devices. In: 2015 IEEE/ACIS 14th International Conference on Computer and Information Science; Las Vegas, NV, USA; 2015. pp. 437-442.

[10] Mikhaylov K, Hänninen T. Mechanisms for improving throughput and energy efficiency of Bluetooth Low Energy for multi node environment. Journal of High Speed Networks 2015; 21 (3): 165-180. doi: 10.3233/JHS-150518

[11] Bluetooth Special Interest Group. Bluetooth Core Specification v 5.0. Bluetooth SIG, 2016.

[12] Collotta M, Pau G, Talty T, Tonguz OK. Bluetooth 5: A concrete step forward toward the IoT. IEEE Communications Magazine 2018; 56: 125-131. doi: 10.1109/MCOM.2018.1700053

[13] Gomez C, Oller J, Paradells J. Overview and evaluation of Bluetooth Low Energy: an emerging low-power wireless technology. Sensors 2012; 12 (9): 11734-11753. doi: 10.3390/s120911734

[14] Cho K, Park W, Hong M, Park G, Cho W et al. Analysis of latency performance of Bluetooth low energy (BLE) networks. Sensors 2015; 15 (1): 59-78. doi: 10.3390/s150100059

[15] Gupta S, Dham R. Improving security with Bluetooth Low Energy 4.2. Embedded Systems Engineering: IoT Security 2016.

[16] Dingle NJ, Knottenbelt WJ, Suto T. PIPE2: a tool for the performance evaluation of generalised stochastic Petri Nets. ACM SIGMETRICS Performance Evaluation Review 2009; 36 (4): 34-39. doi: 10.1145/1530873.1530881

[17] Fafoutis X, Vafeas A, Janko B, Sherratt RS, Pope J et al. Designing wearable sensing platforms for healthcare in a residential environment. EAI Endorsed Transactions on Pervasive Health and Technology 2017; 3 (12): 1-11. doi: 10.4108/eai.7-9-2017.153063