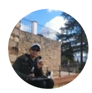
A Synergistic Architecture for RPL over BLE

Taeseop Lee
Seoul National University



Hyung-sin Kim
Seoul National University

A Synergistic Architecture for RPL over BLE

Taeseop Lee, Myung-Sup Lee, Hyung-Sin Kim, and Saewoong Bahk
Department of Electrical and Computer Engineering and INMC
Seoul National University, Seoul, Republic of Korea
E-mail: {tslee, mslee, hskim}@netlab.snu.ac.kr, sbahk@snu.ac.kr

***Abstract*—In this paper, we consider a protocol architecture that enables IPv6 routing protocol for low power and lossy networks (RPL) to run on top of Bluetooth Low Energy (BLE), aiming to provide a BLE-based multi-hop IoT network. In our approach to *RPL over BLE*, we propose to use both of advertising and data channels of BLE to create synergistic effects between RPL and BLE to jointly achieve high energy efficiency and reliable multi-hop routing. We design an adaptation layer between BLE and RPL (*ALBER*) which tightly couples RPL and BLE operations together. Specifically, *ALBER* provides RPL control message broadcast through BLE, RPL routing metric calculation that reflects BLE link quality, and routing table update that incorporates BLE connection management. We implement *ALBER* in a Linux kernel to realize *RPL over BLE* and compare its performance with that of *RPL over IEEE 802.15.4* on a multi-hop testbed network. The performance results show that our architecture is not only feasible but also provides almost perfect packet delivery performance (∼100%) and reduces duty-cycle up to 32% compared to *RPL over IEEE 802.15.4* under varying link dynamics. Our research shows that *RPL over BLE* is a promising approach which can increase the utility and impact of BLE across different application domains.**

***Index Terms*—Low-power Lossy Network (LLN), BLE, RPL, 6LoWPAN, Internet of Things (IoT), IPv6, multi-hop routing**

I. Introduction

Internet of Things (IoT) is a technical megatrend in academia and industry that aims to provide Internet connectivity to resource constrained embedded devices deployed over large areas. When combined with recently advanced technologies such as low power and lossy network (LLN), big data analysis, low cost sensing, and machine learning, IoT has the potential to facilitate a variety of useful applications that impact industry and global markets.

There have been various standardization efforts by IEEE, IETF, ZigBee aliance, and Bluetooth Special Interest Group (SIG) to develop protocols and application profiles for IoT systems. IPv6 routing protocol for LLN, termed RPL [1], is one of such efforts. To support upcoming smart grids and many other IoT applications, RPL constructs a multi-hop LLN with a tree-like routing topology called destination-oriented directed acyclic graph (DODAG), and enables bi-directional IPv6 communication between resource constrained embedded devices.

At the link layer, Bluetooth Low Energy (BLE) and IEEE 802.15.4 are two popular low power and low cost wireless protocols. To date, research (mostly wireless sensor networks) has focused on developing various communication and network techniques on top of IEEE 802.15.4 [2][3][4][5][6][7][8][9]. Recently, IEEE 802.15.4 started to deliver IPv6 packets using 6LoWPAN as an adaptation layer, which provides interoperability with the larger Internet. As a result, IEEE 802.15.4 is widely used in RPL-based multi-hop IoT networks (e.g., Cisco's CG-Mesh network for smart grid [10]) as an underlying link layer protocol. On the other hand, BLE was designed to support single hop range applications which include wearable technologies such as smart watches and some applications in the health care IT.

BLE has not been a focus of wireless sensor network research for multi-hop communication, in contrast to other link layer technologies such as IEEE 802.15.4. In this paper, we aim to show that BLE multi-hop communication can be valuable for several reasons. First, compared to IEEE 802.15.4 [11], BLE provides four times higher data rate (250 kbps vs. 1 Mbps) as well as lower power consumption for transmission and reception. It has the capability of avoiding wireless interference from other devices by exploiting frequency hopping. Second, the synchronous characteristic of its MAC protocol can improve data delivery performance over asynchronous MACs such as ContikiMAC and BoX-MAC, which are widely used for IEEE 802.15.4-based multi-hop networks. Third, from a practical point of view, many of today's smartphones are equipped with BLE which facilitates legacy compatible deployment, accessibility and usability [12].

The Bluetooth Special Interest Group (SIG) defined a new logical link layer channel dedicated to IP-based connections in the Bluetooth 4.1 specification [13]. This specification allows a BLE node to perform multiple roles (master and slave) simultaneously which enables a BLE-based multi-hop network with hierarchical tree topology. In the most recent 4.2 specification, Internet Protocol Support Profiles (IPSP) and HTTP Proxy Service (HPS) have been standardized for the Internet accessibility of Bluetooth smart sensors through a BLE gateway [14][15]. In addition, IETF recently standardized 6LoWPAN for BLE, which enables BLE to deliver IPv6 packets [16].

As part of these trends, in this paper, we consider a protocol stack that aims to run RPL of the IETF standard over BLE to construct BLE-based multi-hop networks. Our motivation is that, when considering the current status and strength of BLE, mounting a standard routing protocol such as RPL on top of BLE may greatly improve its usability and deliver

Saewoong Bahk is the corresponding author.

performance on par with, or better than, IEEE 802.15.4 in large scale applications.

The contributions of this paper are four-fold.

- We interconnect standard routing protocol RPL completely with BLE as a new approach. We propose a synergistic architecture for *RPL over BLE* that constructs an energy-efficient and reliable multi-hop network through connection-less broadcast of RPL control messages and connection-oriented data delivery.
- We design an adaptation layer between BLE and RPL, termed *ALBER*, which effects energy-efficient delivery of RPL control messages, connection-aware parent change, and calculation of BLE-specific routing metric for RPL. We implement the full protocol stack of *RPL over BLE* including *ALBER* in a Linux kernel.
- In an indoor multi-hop LLN testbed, we experimentally evaluate *ALBER*. We show that it enables RPL to run on top of BLE without losing each protocol's strengths, as a result of which *RPL over BLE* significantly outperforms *RPL over IEEE 802.15.4*.
- We experimentally investigate the effect of varying connection interval on the performance of *RPL over BLE* which reveals that connection interval control has a significant effect on *RPL over BLE*'s performance.

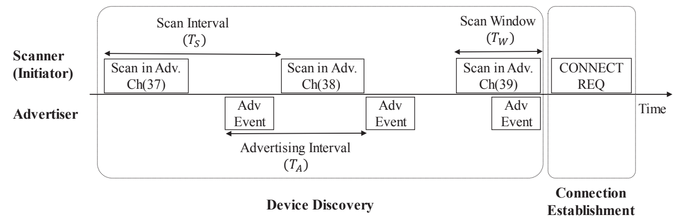
The rest of this paper is organized as follows. Section II provides background on RPL, BLE, and 6LoWPAN for BLE. Section III presents our design of *RPL over BLE* including *ALBER*. Section IV describes its implementation details. In Section V, we evaluate extensively *RPL over BLE*'s performance in an indoor testbed and compare the results with that of *RPL over IEEE 802.15.4*. Section VI summarizes related work.

II. Background

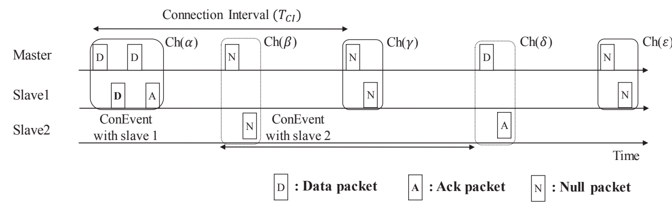
A. RPL operation

In this part, we describe ContikiRPL, which is the default RPL implementation in Contiki. RPL defines *RANK* to represent the routing distance from a node to the root. By default, each node in ContikiRPL uses the end-to-end expected transmission count (*ETX*) toward the root as its own *RANK*. Let $ETX(n, p_n)$ represent the ETX from node n to its parent candidate p_n. Then node n calculates its routing metric for each parent candidate as $R(p_n) = RANK(p_n) + ETX(n, p_n)$. RPL constructs a DODAG by designing each node to select a parent node that has the smallest routing metric among its parent candidate nodes.

Each node broadcasts the routing information including its *RANK* using DODAG information object (DIO) messages. To achieve a balance between control overhead and fast recovery, RPL triggers DIO message transmissions based on the $TrickleTimer$ [17] which doubles the broadcast period after every DIO transmission and re-initializes it to the minimum value when route inconsistency is detected. Each node recognizes its neighbor nodes and their *RANKs* from received DIO messages. RPL constructs downward routes simply as the reverse of upward ones. To this end, each node periodically transmits destination advertisement object (DAO) messages toward the root through its upward route.



(a) Connection-less operation on advertising channels



(b) Connection-oriented operation on data channels

Fig. 1. Link layer operation of BLE in each channel type.

B. BLE link layer operation

BLE has two types of channels: 3 advertising channels and 37 data channels. Advertising channels support connection-less message exchanges through advertising and scanning as depicted in Fig. 1(a), where the advertiser and the scanner denote a sender and a receiver, respectively. The advertiser performs an advertising event (i.e., AdvEvent in Fig. 1(a)) periodically with advertising interval T_A, which comprises three message transmissions, i.e., one for each of the three advertising channels. On the other hand, the scanner tries to receive advertised packets by waking up periodically with scan interval T_S and scanning an advertising channel during a scan window T_W. It scans a different advertising channel at each wake-up.

Based on connection-less message exchanges, BLE performs device discovery and connection establishment procedures in advertising channels. Specifically, each BLE node notifies its presence to its neighbor nodes by advertising control packets. Once a node (scanner) willing to make a connection with the advertiser detects its presence by scanning, it establishes a connection with it by sending a *Connection Request* packet (i.e., CONNECT REQ in Fig. 1(a)). After the two nodes established a connection, each of them becomes a master or slave node according to its role in device discovery (i.e., scanner or advertiser, respectively). Then, they exchange data packets over data channels.

There are two advantages of using data channels rather than advertising channels for data communication. First, the two nodes in connection are synchronized in time domain and share a periodic wake-up schedule (i.e., connection interval T_{CI} in Fig. 1(b)). Every T_{CI} they start a connection event (ConEvent) and exchange null packets to check validity of link connection (shown as 'ConEvent with slave 2' in the figure). If they have data packets for transmission, they exchange them instead of null packets. After the packet exchange is finished, the two nodes close ConEvent and sleep together. Owing to time synchronization, a parent node (master) can schedule its children nodes (slaves) to transmit without overlapping with each other in time domain, thus resulting in contention-free data transmission.

Second, for efficient use of 37 data channels, a parent node and its connected children nodes share a *pseudo-random* hopping sequence that helps to avoid collision with other parent-children (master-slave) pairs. This frequency hopping mechanism is one of its most unique features of BLE that diffentiate it from IEEE 802.15.4. BLE adjusts the hopping sequence considering the link quality of each data channel, which mitigates interference from other systems such as WiFi by removing bad channels from the channel list for the given hopping sequence.

C. 6LoWPAN for BLE

Since RPL is an *IPv6* routing protocol for LLNs which consist of *low cost devices*, a fundamental assumption behind this is that (long and heavy) IPv6 packets have to be delivered through a lightweight link layer. When RPL was introduced, IEEE 802.15.4 was the only lightweight link layer protocol that met this requirement. Specifically, IETF standardized an adaptation layer, named 6LoWPAN [18], which enables IPv6 packet transmission over IEEE 802.15.4 by using header compression and packet fragmentation. This is one of the reasons why existing RPL implementations are built on top of IEEE 802.15.4.

Recently, IETF standardized 6LoWPAN for BLE which opens the possibility of adopting another underlying link layer for RPL. Although 6LoWPAN over BLE provides almost the same functions as that over IEEE 802.15.4, it has some unique features related with BLE connection management. For instance, 6LoWPAN over BLE allows a BLE node to exchange IPv6 packets only with connected nodes and over data channels. Thus, when designing *RPL over BLE*, connection-oriented IPv6 packet exchanges over data channels are necessary, not only to improve performance but also to make it standard-compliant.

III. Design of RPL over BLE

Our work focuses on designing a feasible BLE-based multi-hop mesh network which has great compatibility, extendability, and applicability as part of IoT. To make it IPv6- and standard-compliant, we use standard IP protocols such as RPL and 6LoWPAN for BLE. Our design aims to run RPL over BLE without losing each one's strengths that are RPL's route management and BLE's energy-efficient and reliable packet delivery. To achieve this goal, we consider a new architecture for *RPL over BLE* that utilizes synergistic effects between RPL and BLE, and design *ALBER* to support this architecture. In this section, we describe design details of our proposal.

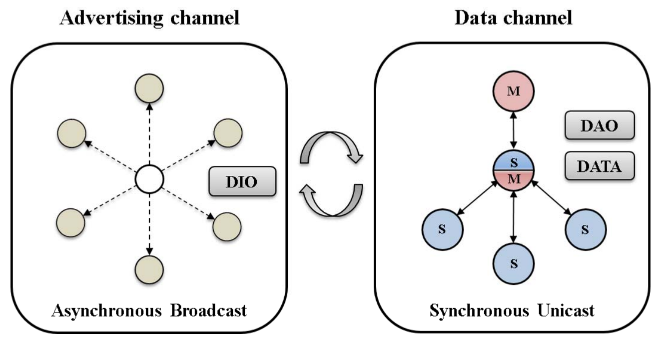


Fig. 2. Design overview of the proposed architecture for RPL over BLE.

A. Synergistic Network Architecture for RPL over BLE

We propose a new network architecture for *RPL over BLE*, which exploits both connection-less and connection-oriented packet exchanges provided by BLE. To this end, we first analyze *pros* and *cons* of using each of the two channel types in BLE and design *RPL over BLE* to use each of them for transmission of different types of packets. The data packet exchange over data channels, which needs to operate with 6LoWPAN, consumes low energy and provides high reliability owing to synchronous operation and channel hopping. However, BLE does not allow a node pair to use data channels before they are connected and synchronized in time domain. This leads BLE to use advertising channels for device discovery and connection establishment procedures that can be performed between two nodes without connection.

We combine channel usage features of BLE with neighbor discovery and parent selection functionalities of RPL, as depicted in Fig. 2. That is, a node broadcasts DIO messages over advertising channels, which enables its unconnected neighbor nodes to obtain its routing information. Each node selects a parent node among its parent candidates using received DIOs from its neighbor nodes, and then establishes a BLE connection with a parent node through advertising channels. Then, it exchanges data packets and DAOs with the connected parent node (master) over data channels.

Even after a node joins the network through a valid parent node, it scans advertising channels every scan interval to update its parent candidate set by using received DIOs. Furthermore, it performs periodic wake-ups in data channels and exchanges data packets with its parent node or a child node. This design choice allows resource constrained nodes to benefit from both RPL and BLE. That is, they exchange data packets reliably and energy efficiently using BLE, while constructing and recovering multi-hop routing topology using RPL.

Lastly, this architecture completely avoids collision between control and data packets because their transmissions are separated in the frequency domain, which provides contention-free data delivery even when the routing topology dynamically changes.

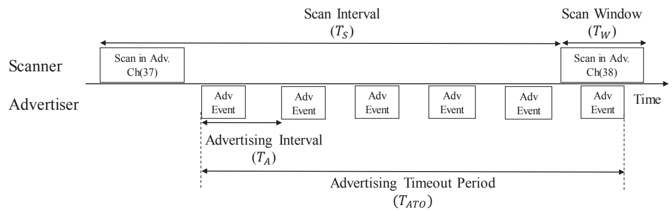
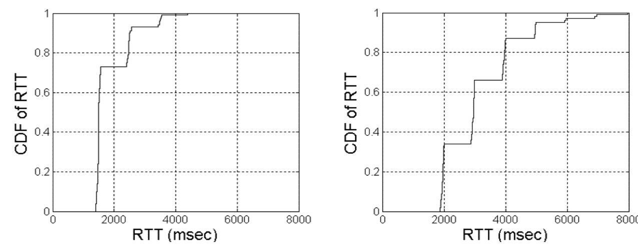


Fig. 3. Exchange of RPL control messages over advertising channels with low power listening.



(a) LOS link with 10 m distance (b) NLOS link with 10 m distance

Fig. 4. CDF of Logical Link layer ping's RTT with $T_{CI} = 1sec$.

B. DIO broadcast over advertising channels

In the BLE standard specification, unlike data channels, advertising channels deliver packets in an asynchronous manner, which is highly affected by the four operation parameters: scan interval (T_S), scan window (T_W), advertising timeout period (T_{ATO})[1], and advertising interval (T_A). However, when using advertising channels with the default parameter settings, a BLE node suffers from low packet delivery ratio and low energy efficiency. We need to improve the performance by tuning these parameters for routing topology management, given that DIO messages that contain important routing information are delivered through advertising channels.

We use a low power listening (LPL) protocol [3] with proper settings of the above four parameters, as depicted in Fig. 3. For reliable delivery in advertising channels, a scanner detects at least one advertising packet during a scanning procedure, which runs with the two constraints: $T_{ATO} \geq T_S$ and $T_A \leq T_W$. In addition, the transmission overhead needs to be minimized to reduce energy consumption for sending advertising packets. To this end, we set $T_{ATO} = T_S$ and $T_A = T_W$.

With this configuration, we further minimize energy consumption of a node in *advertising* channels by parameter optimization. The average power consumption of a BLE node can be expressed as

$$P = \left(\frac{T_W}{T_S}\right) I_s V + \left(\frac{3T_a T_{ATO}}{T_A} \times \frac{1}{T_{DIO}}\right) I_a V \tag{1}$$

where V is the voltage of power supply, and I_s and I_a are the current consumptions during scanning and advertising, respectively. T_{DIO} represents the DIO broadcast interval and T_a is the packet transmission time. Here, the left term comes from the fact that a node scans an advertising channel for T_W every T_S, which results in the default duty-cycle of T_W/T_S. The right term represents energy consumption for message transmission. An advertising node generates an advertising event T_{ATO}/T_A times for a DIO message delivery. In addition, each advertising event has three packet transmissions over the three advertising channels, which results in the duty-cycle of $(3T_a \cdot T_{ATO})/(T_A \cdot T_{DIO})$ for DIO transmission.

[1]For DIO message broadcasting, a BLE node repeats advertising events during an advertising timeout period. Even though BLE specification does not specify this period, we newly define it for our design, given that a BLE chipset allows its upper layer to start or stop sending advertising packets.

Using the two conditions of $T_{ATO} = T_S$ and $T_A = T_W$, and letting $X = T_W/T_S$, we can rewrite Eq. (1) as

$$P = I_s V X + \frac{3T_a I_a V/T_{DIO}}{X}. \tag{2}$$

Then we can easily get an optimal value of X that minimizes the power consumption as

$$X_{opt} = \sqrt{\frac{3T_a I_a}{I_s T_{DIO}}}. \tag{3}$$

Given that the current consumptions of a BLE node during receiving and transmitting modes are almost the same (i.e., $I_a \simeq I_s$) [19], we can further simplify X_{opt} as $\sqrt{3T_a/T_{DIO}}$. Using this value, the LPL protocol can deliver DIO messages over advertising channels with minimal energy consumption.

C. Routing metric for RPL over BLE

Although RPL decouples routing metric from the standard specification, the most widely used routing metrics are end-to-end ETX and the combination of ETX and hop distance. However, these routing metrics cannot be applied to *RPL over BLE* since BLE link layer does not provide information about link layer retransmission for the upper layer. To help link quality estimation that is necessary to obtain routing metric, *ALBER* makes each child node periodically transmit a ping packet of Logical Link Control and Adaptation Protocol (L2CAP) to its parent node over data channels, and uses its round trip time (RTT).

Fig. 4 shows the CDF of RTT values for link layer ping packets under two different link conditions (line-of-sight (LOS) and non-LOS (NLOS) links with the distance of 10 m) when the connection interval is 1 second. We use this figure to investigate feasibility of using an RTT-based link quality metric and get an intuition about how to obtain a useful routing metric. We first observe that RTT values in the LOS link condition is much smaller than those in the NLOS link condition. This reveals that RTT reflects link quality indeed and has potential to be used as a link quality metric. Furthermore, we find out that RTT values are clearly quantized with the step size of connection interval, which comes from BLE's retransmission strategy. That is, when a BLE node fails to transmit a packet, it closes its current connection event and delays its retransmission attempt until the next connection

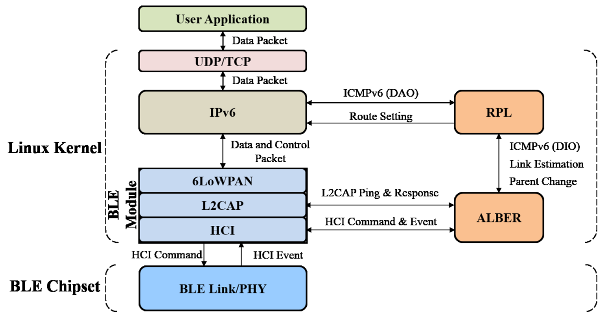


Fig. 5. Protocol stack of *RPL over BLE* including *ALBER*.

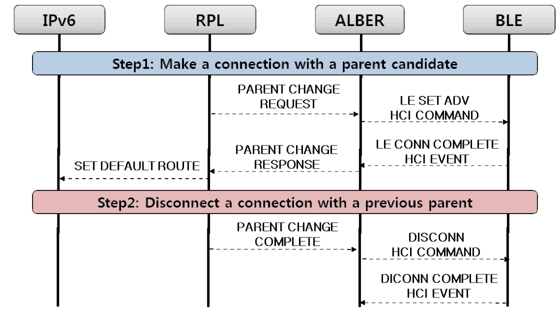


Fig. 6. Connection-aware parent change procedures in *RPL over BLE*.

event starts. This leads each retransmission to be delayed by one connection interval.

From these observations, we can simplify each measured RTT value as the number of connection intervals required for a packet transmission, i.e.,

$$N_{CI} = \left\lfloor \frac{RTT}{T_{CI}} \right\rfloor . \tag{4}$$

Then, we define a new metric, called ECI (Expected number of Connection Interval), as the exponentially weighted moving average of N_{CI}, and use it to represent link quality. Accordingly, node n calculates its routing metric for its parent candidate p_n as $R(p_n) = RANK(p_n) + ECI(n, p_n)$ where $RANK(p_n)$ is the end-to-end ECI between p_n and the root, and $ECI(n, p_n)$ is the ECI between node n and its parent p_n.

Our link estimation scheme has a trade-off relation between ping packet overhead and estimation accuracy. In this sense, we find out that the additional overhead is marginal compared to that of BLE's default connection management mechanism. Since a connected node sends a ping packet (21 bytes) instead of a null packet (10 bytes), each ping exchange requires only 176 μsec more radio-on time due to its 11 bytes larger packet size. In our testbed experiments, we empirically set the ping packet interval as 10 seconds, which results in only a 0.00176% increase in the duty cycle. Our performance evaluation shows that this ping packet interval is small enough to change routing topology according to link conditions, and our link estimation scheme is suitable to *RPL over BLE*.

D. RPL parent change with BLE connection management

Fig. 5 shows the protocol stack of *RPL over BLE* that includes *ALBER*. Differently from *RPL over IEEE 802.15.4*, *RPL over BLE* allows a node to exchange IPv6 data packets through 6LoWPAN only with connected nodes. If the RPL of a node changes its parent node arbitrarily, 6LoWPAN starts to drop all the packets targeted at a new parent because it cannot find a valid connection with the new parent node.

To reduce inefficient packet losses, we design a seamless parent change mechanism for *RPL over BLE* as illustrated in Fig. 6, which allows *ALBER* (rather than RPL) to determine whether to change a parent node considering BLE's connection status. Specifically, when RPL needs to select a new parent node, it notifies this event to *ALBER*, rather than updating the routing table hastily. Then, *ALBER* makes BLE link layer establish a connection with the new parent node. After BLE link layer creates a connection, it reports *Connection Complete Event* to *ALBER*, which enables RPL to change the default route entry to the new parent node. When the BLE link layer fails to make a connection, RPL selects another parent node and repeats the same procedures again. Finally, after *ALBER* is reported that RPL finished updating the routing table, it makes BLE link layer cut off the connection with the old parent node and terminates the parent change procedures.

TABLE I
HCI COMMANDS FOR EACH OPERATION OF *RPL over BLE*

Operation	HCI commands
RPL control message exchange	-LE Set Scan Parameter Command -LE Set Scan Enable Command -LE Set Advertising Parameter Command -LE Set Advertising Data Command -LE Set Advertise Enable Command
Parent change with connection management	-LE Create Connection Command -LE Set Advertising Parameter Command -LE Set Advertise Enable Command -Disconnect Command

IV. *ALBER* IMPLEMENTATION

We have implemented RPL over BLE in Linux kernel version 3.17 with the use of a BLE commercial chipset and the BLE module covering HCI, L2CAP and 6LoWPAN, which are available in the current kernel version. We first implemented RPL in the Linux kernel based on ContikiRPL source code [20], and placed *ALBER* between RPL and the BLE module, as depicted in Fig. 5. This hierarchical structure is designed for *ALBER* to perform required functions, and provides backward compatibility while minimizing modification of other layers.

Even though BLE's link and PHY layers (controller part) are generally implemented on a BLE hardware chipset, BLE provides various Host Controller Interface (HCI) commands, which helps the upper layer (host part) to manage some link level operations and determine lower layer parameters.

We design *ALBER* to use some of these HCI commands to interconnect RPL with BLE, as summarized in Table I. Then, the HCI layer controls BLE's link layer operation when it receives HCI commands from *ALBER*. Moreover, it allows *ALBER* to have necessary link level information for the operation of *RPL over BLE* by reporting some link layer

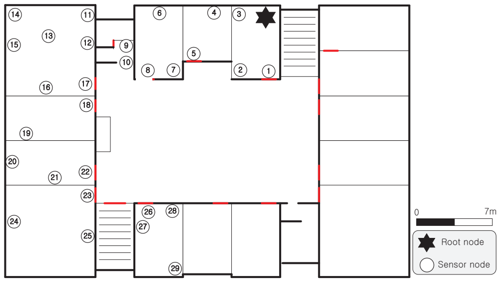
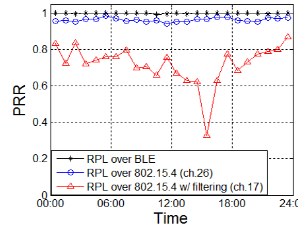
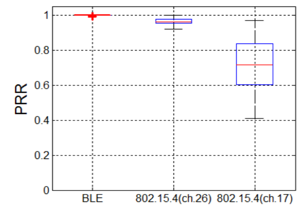


Fig. 7. Testbed topology map with 30 nodes and one root.

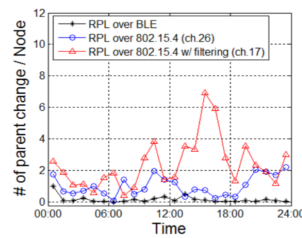
Fig. 8. TelosB for IEEE 802.15.4 node (left), a set of Raspberry Pi and BCM4356 for BLE node (middle), and NI USB-6210 for power meter (right).



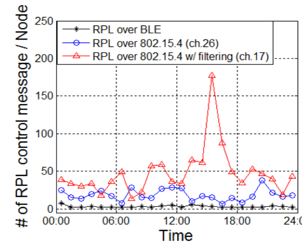
(a) PRR vs. time



(b) PRR of each node



(c) Parent changes vs. time



(d) Routing overhead vs. time

Fig. 9. Performance metrics of *RPL over BLE* and *RPL over IEEE 802.15.4* for 24 hours in a light traffic scenario (5 minutes/packet/node).

events such as connection establishment and advertising packet reception. Finally, *ALBER* sends an L2CAP ping request to the L2CAP layer, and updates its ECI value by the received RTT value.

We modify ContikiRPL to support our *RPL over BLE* as follows. First, we make RPL send DIO messages through *ALBER* which achieves connection-less DIO message delivery over *advertising* channels[2]. Second, we design RPL to update its routing table after having *ALBER*'s approval. Finally, we use ECI (instead of ETX) as a routing metric to represent BLE's link quality.

V. Performance Evaluation

A. Testbed environments

We configured a testbed topology in an indoor office environment as depicted in Fig. 7, where a total of 31 nodes were deployed with one root node (marked with the star). For an IEEE 802.15.4 node, we use a TelosB clone device [21] with an MSP430 microcontroller and a CC2420 radio, and use a transmission power of 0 dBm with an antenna gain of 5 dB. For a BLE node, we use a Raspberry Pi device with Linux kernel version 3.17 and a Broadcom BCM4356 chipset as a BLE radio, with the same antenna gain of 5 dB as for an 802.15.4 node. Lastly, we use NI USB-6210 to measure the duty-cycle of a BLE radio since the BLE chipset does not provide this information. Fig. 8 shows the devices we used in our experiments.

With the above hardware setup, we use a set of ContikiRPL and ContikiMAC of Contiki 2.7 to evaluate the performance of *RPL over IEEE 802.15.4*, where the default sleep interval of ContikiMAC is 125 msec. For *RPL over BLE*, we use the connection interval of 50 msec (i.e., default value in HCI layer)

[2]Note that DAOs and data packets in our design are delivered through IPv6 and 6LoWPAN.

and determine the parameters for scanning and advertising operations on *advertising* channels based on the optimization in Section III-B. Although our mathematical analysis gives BLE's default duty-cycle (T_W/T_S) of only 0.1%, we practically exploit a sightly larger value of 0.2% in our experiments since the current BLE specification gives the parameter range requirements as $T_A(=T_W) \geq$20 msec and $T_S \leq$10.24 sec.

B. Comparison of RPL over BLE vs. RPL over 802.15.4

We first evaluate the effect of link dynamics on the performance of *RPL over BLE* by delivering a light upstream traffic[3] (5 minutes/packet/node) for 24 hours. For comparison, we also evaluate the performance of *RPL over IEEE 802.15.4* on both channels of 17 and 26[4].

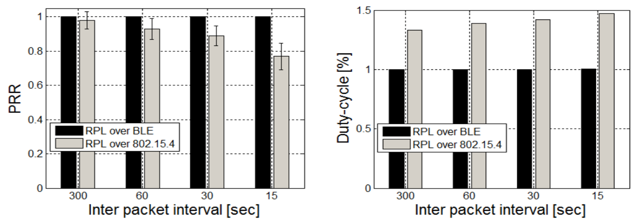
Figs. 9(a) through 9(d) plot various performance metrics per hour. We first observed that *RPL over IEEE 802.15.4* on channel 17 performs too bad (i.e., packet reception ratio (PRR) of $\sim$0%) during a day-time due to WiFi interference. Thus, we plot the results of another case that a node selects a parent node only when its received signal strength is higher than -70 dBm. Fig. 9(a) shows that this filtering-based parent selection improves PRR performance on channel 17. However, it still shows the worst PRR all the time, and this PRR performance was further degraded during a day-time as low as $\sim$25%. This reveals that *RPL over IEEE 802.15.4* significantly suffers from WiFi interference.

On channel 26, *RPL over IEEE 802.15.4* provides much better PRR. However, it still experiences some level of PRR degradation since ContikiMAC detects on-going packet transmissions using energy detection for an extremely short period[5],

[3]For brevity, we omitted the performance evaluation of downstream traffic delivery in this paper.

[4]Here channel 17 is with much heavier WiFi interference than channel 26.

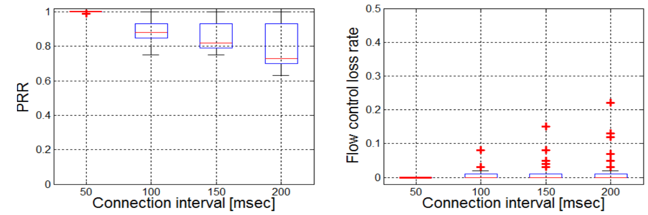
[5]This design choice minimizes energy consumption for periodic wake-ups.



(a) PRR vs. traffic load

(b) Duty-cycle of a leaf node (node 30) vs. traffic load

Fig. 10. Performance comparison of *RPL over BLE* vs. *RPL over IEEE 802.15.4* with varying traffic load.



(a) PRR of each node vs. connection interval

(b) Queue loss ratio of each node vs. connection interval

Fig. 11. Performance of *RPL over BLE* with varying connection interval in a heavy traffic scenario (30 seconds/packet/node).

which incurs packet losses due to its low probability of energy detection.

On the other hand, *RPL over BLE* provides not only the best performance among competitors, but also almost perfect PRR performance all the time. This is because BLE's frequency hopping mechanism enables *RPL over BLE* to overcome link dynamics and WiFi interference. Moreover, owing to use of a synchronous MAC, a BLE node knows when its connected nodes wake up and send packets based on time synchronization, i.e., not energy detection. Lastly, Fig. 9(b) reveals that *RPL over BLE* provides nearly perfect PRR fairness among the nodes while *RPL over IEEE 802.15.4* shows node location-dependent PRR due to each node's different link condition.

Figs. 9(c) and 9(d) show that *RPL over IEEE 802.15.4* takes a great effort to overcome link dynamics, which results in frequent parent changes and more routing control packet transmissions. However, the results in Fig. 9(a) reveal that these efforts are not sufficient to reduce packet losses. In contrast, our *RPL over BLE* maintains quite stable routing topology and much lower routing overhead compared to *RPL over IEEE 802.15.4* all the time, which clearly shows a positive effect of BLE's connection-based operation.

In Figs. 10(a) and 10(b), we compare *RPL over BLE* and *RPL over IEEE 802.15.4* (on channel 26) under varying traffic load in terms of PRR and duty-cycle[6]. In these experiments, we make the sleep interval of ContikiMAC equal to the connection interval of BLE (i.e., 50 msec) for fair comparison. As traffic load increases, *RPL over IEEE 802.15.4* experiences significant PRR degradation as low as ∼80%. In ContikiMAC, a sender transmits packets repetitively during a whole sleep interval because it does not know when its receiver wakes up, which incurs contention and throughput degradation at heavy load even without WiFi interference.

On the other hand, *RPL over BLE* maintains nearly perfect PRR regardless of traffic load since it does not require repetitive transmissions. Furthermore, in this architecture, we design a parent node (master) to schedule connection events of its children nodes, and to transmit control and data packets over different channel types, which leads to contention-free data transmissions.

[6]We measured the duty-cycle of a leaf node (node 30) to focus on the effect of link layer operation on energy consumption, rather than on time-varying routing topology.

Fig. 10(b) shows that *RPL over BLE* provides lower duty-cycle than *RPL over IEEE 802.15.4*, and the performance gain further increases up to 32%. This is because BLE's synchronous operation does not require repetitive (and redundant) transmissions, and its higher data rate (1 Mbps) helps to reduce packet transmission time. In addition, ContikiMAC has a false wake-up problem due to its energy-based packet detection and consumes more energy, which is not a problem in the case of BLE.

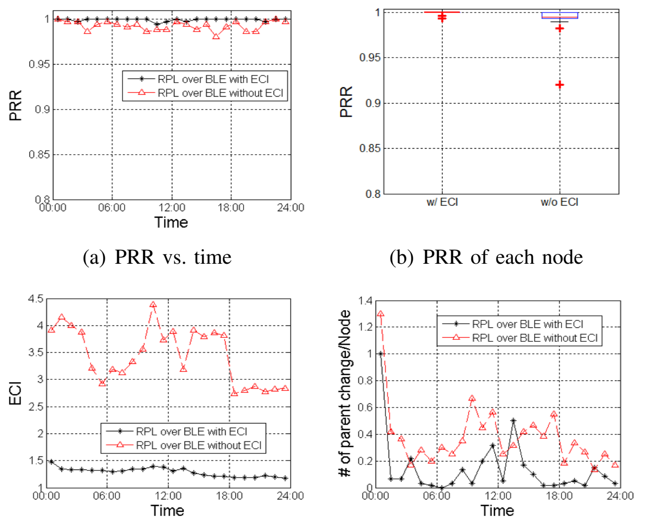
From the results of duty-cycle performance, we can infer three more things. First, given that the duty-cycle gain of *RPL over BLE* increases with traffic load, it benefits bottleneck nodes in a multi-hop network more significantly. Second, given that BLE consumes lower energy than IEEE 802.15.4 for transmission and reception, the energy saving effect of *RPL over BLE* overwhelms its duty-cycle improvement. Finally, given that the total duty-cycle of *RPL over BLE* was measured as 1% and the default duty-cycle (T_W/T_S) in advertising channels is 0.2%, we can infer that connection-based operation on data channels requires duty cycle of 0.8% (four times larger than the default one)[7]. This implies that we may further prolong battery lifetime by increasing connection interval.

C. *Effect of varying connection interval*

Figs. 11(a) and 11(b) depict end-to-end PRR and packet loss ratio at link layer queue of each node, as the connection interval increases from the default value, when the data packet interval is 15 seconds/packet/node. Our results show that PRR decreases with connection interval since each node has less chances to transmit packets to its parent node. Specifically, a node with many children nodes can experience more queue losses since it receives too many packets during a connection interval without having any opportunity to transmit packets. Given that use of a larger connection interval has potential to save energy, we need to carefully determine the connection interval to achieve required throughput with minimal energy consumption.

Furthermore, from Fig. 11(b), we observe that only a few nodes experience packet losses much more severely than other nodes. This unfair packet loss rate comes from the fact that a multi-hop network naturally makes nodes near the root node suffer much more relay burden and RPL has the load balancing

[7]This includes periodic wake-ups and null (or data) packet exchange.



(a) PRR vs. time

(b) PRR of each node

(c) ECI vs. time

(d) Parent change vs. time

Fig. 12. Performance of *RPL over BLE* with and without our ECI-based routing metric, for 24 hours in a light traffic scenario (5 minutes/packet/node).

problem [22]. This gives us an intuition that PRR performance of *RPL over BLE* can be improved by reducing the connection interval only for a few bottleneck nodes, without consuming large energy. From these results, we believe that connection interval control can be a valuable research topic for *RPL over BLE*.

D. Effect of ECI-based routing metric

Now we investigate the effect of using ECI-based routing metric on the performance of *RPL over BLE*. To this end, we use Figs. 12(a) through 12(d) which plot various performance metrics per hour when *RPL over BLE* delivers light traffic (5 minutes/packet/node) for 24 hours with and without the proposed routing metric. Given that a BLE node does not feedback its link quality to the upper layers, we configure '*RPL over BLE* without ECI' to use only hop distance for its routing metric[8].

Fig. 12(a) shows that our ECI-based routing metric for *RPL over BLE* leads to a slightly better average PRR all the time since it reflects link quality by using periodic ping packet transmissions. This can be verified in Fig. 12(c), which shows that the proposed routing metric provides much lower link layer ECI by enabling each node to select a parent node with good link quality. This enhanced parent selection stabilizes routing topology and results in a reduced number of parent changes as shown in Fig. 12(d). Furthermore, Fig. 12(b) reveals that the new routing metric improves PRR fairness among nodes. This confirms necessity of using the ECI-based routing metric by showing that, depending on physical topology, a certain node continuously suffers from packet losses without using link quality for routing metric.

[8]This still measures ECI for performance comparison.

VI. Related Work

A number of studies have investigated the performance of RPL [1] and IEEE 802.15.4 under various network configurations. Ko *et al*. experimentally evaluated the performance of RPL over IEEE 802.15.4 using TinyOS [23] and showed that its performance is similar to that of the widely used collection tree protocol (CTP) [6], while benefiting from an IPv6-based architecture. The work in [24] investigated the performance of TCP over RPL over IEEE 802.15.4, which shows that RPL-based multi-hop networks suffer from throughput unfairness when delivering TCP traffic. Bressan *et al*. studied the performance of RPL by simulation in the context of smart grids [25]. Gungor *et al*. measured IEEE 802.15.4 link qualities in real power grid environments and discussed associated opportunities along with their challenges [26]. However, they did not consider BLE as an underlying link layer protocol.

A number of research efforts have pointed out problems associated with RPL and proposed methods to alleviate them. The work in [22] investigated the load balancing problem of RPL and proposed QU-RPL to solve it using information of queue utilization. Ancillotti *et al*. proposed a cross-layering design for RPL which provides enhanced link estimation and efficient management of neighbor tables [27]. Kim *et al*. proposed an asymmetric transmission power-based network to improve downward packet delivery of RPL in dynamic link environments [28][29]. None of these, however, addressed protocol design and architectural issues for implementing RPL over BLE.

BLE has received significant attention due to its low energy consumption, even compared to IEEE 802.15.4. Siekkinen *et al.* experimentally verified that BLE consumes less energy than IEEE 802.15.4 [30]. Kamath and Lindh reported energy consumption of BLE at each specific operation through a measurement study using BLE radio chips (CC2541) [19]. Liu *et al.* mathematically analyzed the energy consumption of BLE operations for device discovery procedure [31][32]. Philipp *et al.* presented a precise energy model for BLE link layer operation [33]. These studies investigated energy consumption of BLE in single hop connections without consideration of multi-hop networks.

Some recent work looked at constructing a multi-hop BLE network which may be classified into two groups: flooding- vs. connection-based approaches. As an example of the first approach, Cambridge Silicon Radio (CSR) developed a flooding-based BLE mesh network, termed *CSRmesh* [34]. Nordic Semiconductor developed a BLE-based multi-hop mesh network by using a flood control algorithm called *Trickle* [17][35]. Bluetooth SIG organized a smart mesh working group for devising a new specification of a flooding-based BLE mesh network [36]. These efforts represent growing industry interest to invest in BLE-based multi-hop networks and provide an initial evidence of its feasibility. However, flooding-based approaches cannot exploit advantages of BLE's connection-based operation which include channel hopping, synchronous data exchange, and link level retransmission.

In the second group, several studies explored feasibility of a connection-based multi-hop BLE network [37][38][39]. A limitation of these works is that they tested basic operations without providing comprehensive performance evaluation. They also did not consider RPL over BLE issues and their impact of network layer IP packet delivery.

Our research addresses gaps left by the previous works and advances a new connection-oriented multi-hop BLE network through *RPL over BLE* which is IP- and standards-compliant. *RPL over BLE* enables siginficant performance improvement compared to 802.15.4 based approaches by exploiting the strengths and synergism of RPL and BLE.

VII. Conclusion

In this paper, we interconnected RPL with BLE and constructed a BLE-based multi-hop network. Our approach was motivated by the potential that a standard-compliant multi-hop BLE network has great potential in the IoT domain. This is due to BLE being more accessible than IEEE 802.15.4, having a higher data rate compared to IEEE 802.15.4, using frequency hopping to mitigate interference, and synchronous operation.

We proposed a synergistic network architecture that incorporates a redesigned RPL control message broadcast, RPL parent change procedures, and RPL routing metric. This is captured by a new adaptation layer, termed *ALBER*, which enables tight coupling of RPL and BLE operations.

We evaluated the performance of *RPL over BLE* through extensive experiments in an indoor testbed. We compared the performance results with that of *RPL over IEEE 802.15.4* which showed significant improvement. We also showed that *RPL over BLE* is quite robust under varying link dynamics and WiFi interference. The proposed architecture significantly improves end-to-end packet delivery performance with low duty-cycle.

References

[1] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012.

[2] M. Buettner *et al.*, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *Proc. ACM SenSys'06*, Nov. 2006.

[3] D. Moss and P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Bounrdaries in Low-Power Networking," Stanford Information Networks Group, Tech. Rep. SING-08-00, 2008.

[4] P. Dutta *et al.*, "Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless," in *Proc. ACM Sensys'10*, Nov. 2010.

[5] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," SICS Report, Tech. Rep. 5128, 2011.

[6] O. Gnawali *et al.*, "Collection Tree Protocol," in *Proc. ACM SenSys'09*, Nov. 2009.

[7] O. Landsiedel *et al.*, "Low Power, Low Delay: Opportunistic Routing Meets Duty Cycling," in *Proc. IPSN'12*, Apr. 2012.

[8] F. Ferrari *et al.*, "Efficient Network Flooding and Time Synchronization with Glossy," in *Proc. IPSN'11*, Apr. 2011.

[9] F. Ferrari *et al.*, "Low-Power Wireless Bus," in *Proc. ACM SenSys'12*, Nov. 2012.

[10] Cisco, "Connected Grid Networks for Smart Grid - Field Area Network," http://www.cisco.com/web/strategy/energy/field_area_network.html.

[11] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and Simpliciti," *Journal of Sensor and Actuator Networks*, vol. 2, no. 3, pp. 589–613, 2013.

[12] R. Heydon, *Bluetooth Low Energy: The developer's handbook*. Prentice Hall, 2012.

[13] "Bluetooth 4.1 Features & Technical Descriptions," Bluetooth SIG, Nov. 2013, [Online]. Available: http://www.bluetooth.com/.

[14] "Bluetooth Core Specification 4.2 Frequently Asked Questions," Bluetooth SIG, Dec. 2014, [Online]. Available: http://www.bluetooth.com/.

[15] J. Decuir, "Introducing Bluetooth Smart: Part ii: Applications and Updates." *IEEE Consumer Electronics Magazine*, vol. 3, no. 2, pp. 25–29, 2014.

[16] J. Nieminen *et al.*, "IPv6 over Bluetooth Low Energy," RFC 7668, Oct. 2015.

[17] P. Levis, T. Clausen, J. Hui, and O. Gnawali, "The Trickle Algorithm," RFC 6206, Mar. 2011.

[18] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4," RFC 4944, Sep. 2007.

[19] S. Kamath and J. Lindh, "Measuring Bluetooth Low Energy Power Consumption," *Texas instruments application note AN092, Dallas*, 2010.

[20] "Contiki RPL code for Linux," Joo Pedro Taveira, Jan. 2014, [Online]. Available: https://github.com/joaopedrotaveira/linux-rpl.

[21] Moteiv Corporation, "Tmote Sky," Aug. 2013, [Online]. Available: http://wirelesssensornetworks.weebly.com/1/post/2013/08/tmote-sky.html.

[22] H.-S. Kim, J. Paek, and S. Bahk, "QU-RPL: Queue Utilization based RPL for Load Balancing in Large Scale Industrial Applications," in *Proc. IEEE SECON'15*, Jun. 2015.

[23] J. Ko *et al.*, "Evaluating the Performance of RPL and 6LoWPAN in TinyOS," in *Proc. IPSN Workshop*, Apr. 2011.

[24] H.-S. Kim *et al.*, "A Measurement Study of TCP over RPL in Low-Power and Lossy Networks," *Journal of Communications and Networks*, vol. 17, no. 6, Dec. 2015.

[25] N. Bressan *et al.*, "The Deployment of a Smart Monitoring System Using Wireless Sensor and Actuator Networks," in *Proc. IEEE SmartGridComm'10*, Oct. 2010.

[26] V. Gungor, B. Lu, and G. Hancke, "Opportunities and Challenges of Wireless Sensor Networks in Smart Grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.

[27] E. Ancillotti *et al.*, "Reliable Data Delivery with the IETF Routing Protocol for Low-Power and Lossy Networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 3, pp. 1864–1877, 2014.

[28] H.-S. Kim, Y.-J. Choi, and S. Bahk, "Elimination of Multi-Hop Transmission from Downlink in Low Power and Lossy Networks," in *Proc. IEEE ICC'14*, June. 2014.

[29] H.-S. Kim *et al.*, "MarketNet: An Asymmetric Transmission Power-Based Wireless System for Managing e-Price Tags in Markets," in *Proc. ACM SenSys'15*, Nov. 2015.

[30] M. Siekkinen *et al.*, "How Low Energy is Bluetooth Low Energy? Comparative Measurements with Zigbee/802.15.4," in *Proc. IEEE WCNC Workshop*, Apr. 2012.

[31] J. Liu, C. Chen, and Y. Ma, "Modeling Neighbor Discovery in Bluetooth Low Energy Networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1439–1441, 2012.

[32] J. Liu *et al.*, "Energy Analysis of Device Discovery for Bluetooth Low Energy," in *Proc. IEEE VTC Fall*, Sep. 2013.

[33] K. Philipp *et al.*, "Precise Energy Modeling for the Bluetooth Low Energy Protocol," *arXiv preprint arXiv:1403.2919*, 2014.

[34] "CSRmesh," Cambride Silicon Radio (CSR), 2015, [Online]. Available: https://wiki.csr.com/wiki/CSRmesh.

[35] "nRF51-ble-broadcast-mesh," NordicSemiconductor, 2015, [Online]. Available: https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh.

[36] K.-H. Chang, "Bluetooth: A Viable Solution for IoT? [Industry Perspectives]," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, 2014.

[37] K. Mikhaylov and J. Tervonen, "Multihop Data Transfer Service for Bluetooth Low Energy," in *Proc. IEEE ITST'13*, Nov. 2013.

[38] B. K. Maharjan, U. Witkowski, and R. Zandian, "Tree Network Based on Bluetooth 4.0 for Wireless Sensor Network Applications," in *Proc. IEEE EDERC'14*, Sep. 2014.

[39] Z. Guo *et al.*, "An On-demand Scatternet Formation and Multi-hop Routing Protocol for BLE-based Wireless Sensor Networks," in *Proc. IEEE WCNC'15*, Mar. 2015.