# Wi-BLE: On Cooperative Operation of Wi-Fi and Bluetooth Low Energy under IPv6

Myungsup Lee, Taeseop Lee, Hyung-Sin Kim, and Saewoong Bahk

*Abstract*—**Wi-Fi and Bluetooth Low Energy (BLE) are popular wireless communication techniques and with the development of integrated combo chips, it is common these days that a single mobile/edge/IoT device supports both Wi-Fi and BLE. Given that the two wireless techniques have clearly different *pros* and *cons* (e.g., high throughput vs. low power consumption), a device can experience significant performance improvement if the two link layers are utilized adaptively according to wireless environments. However, since it is a nontrivial challenge to address their heterogeneity, the two have been used for different purposes as separated silos. In this paper, we investigate how to operate Wi-Fi and BLE synergistically under IPv6 in the context of a low-power multihop network. Specifically, we design *Wi-BLE* , an interface between IPv6 and link layer that handles both control plane (multihop route construction) and data plane (link layer choice for data transmission) for IPv6 over Wi-Fi and BLE. In doing so, we consider various aspects of the two wireless links, such as throughput, transmission range, and power consumption. In addition, to compensate for the lack of a low-power routing protocol for BLE, we design MABLE, a submodule of *Wi-BLE*, that builds ad hoc routes energy-efficiently by deeply considering BLE's connection-based operation. We have implemented *Wi-BLE* on the Linux kernel and evaluated its performance on an indoor testbed. Our empirical results verify that the cooperative use of Wi-Fi and BLE improves reliability and energy efficiency significantly.**

*Index Terms*—**BLE, IPv6, Wi-Fi, wireless communication.**

## I. INTRODUCTION

**W**IRELESS connectivity is an essential part of many useful applications in our daily living. Each application has different requirements for wireless connectivity and accordingly, a variety of wireless radios, such as Wi-Fi, LTE, Bluetooth, LoRa, UWB, and IEEE 802.15.4, have been developed to satisfy these requirements. Each radio, designed for its own purpose, has different characteristics in terms of transmission range, data rate, power consumption, frequency

band, and cost, for example, resulting in different *pros* and *cons*.

As Internet of things (IoT) technology grows and its applications are diversified, however, it is not practical to assume that users may purchase a different device for each of these diverse applications; a single device is expected to support multiple different applications for convenience. To this end, it is common that smart devices such as smartphones and wearables have multiple types of radios and even various combo chips where different radios are integrated are now off the shelf. Given the widely used multi-radio hardware, a question naturally arises: "*Can we operate these multiple different radios in a single device synergistically to meet various application needs?*"

To answer the question, we investigate collaborative operation of Wi-Fi and Bluetooth Low Energy (BLE) that are most widely used in IoT applications and commonly equipped on a single IoT device. While integrated into a single combo chip, Wi-Fi and BLE have quite different characteristics in terms of transmission range, data rate, and power consumption, and they are used separately for different applications as silos. Wi-Fi has a relatively wide transmission range and high data rate, while it consumes high power, resulting in significantly reduced life time of IoT devices. On the other hand, BLE consumes very low energy, making it suitable for battery-constrained IoT devices, but it cannot support high data rate. Both high data rate and long life time are important requirements for IoT, but previous studies have focused on only one aspect of the two. In particular, there has been a lack of research on the energy-efficient use of radio combinations with different characteristics while fully utilizing all the features of the combinations. In this work, we aim to find a *sweet spot* of the trade-off in the context of multihop IoT networks. In other words, our goal is to achieve **high energy efficiency while providing throughput** required by an application.

**Challenges:** To this end, there are a set of non-trivial challenges to be addressed both in control and data planes. (1) In the control plane, routing over Wi-Fi and BLE creates multihop routes, each with a different hop distance and control overhead. To verify the synergy between Wi-Fi and BLE, using the two radios together should end up constructing reliable and efficient multihop routes with low control overhead. For example, for control message transmission, what radio to use and when to use it should be investigated carefully. (2) In addition to collaborative route construction, ad hoc routing for BLE should be newly designed to keep its low-power characteristic in multihop networks. Existing routing protocols, such as AODV [1] and even BLEmesh [2], do not

consider BLE's connection-based link layer operation, wasting a significant amount of energy with advertising-based links. (3) In the data plane, a radio interface for data transmission should be selected in a way that energy consumption is minimized while an application's throughput requirement is satisfied. It is challenging because a radio interface choice causes differences in not only data rate and power consumption but also hop distance, all of which impact both energy consumption and application-level throughput.

**Approach:** We design *Wi-BLE*[1] to resolve the challenges above while being compatible to IPv6 as part of "I"oT networks, because IPv6 is indispensable for IoT networks in terms of security, scalability, and connectivity. Specifically, we investigate three options to enable collaborative routing using Wi-Fi and BLE: (1) Wi-Fi-based routing with BLE wake-up radio, (2) BLE-based routing, and (3) BLE-based routing and Wi-Fi-based route optimization, analyzing their *pros* and *cons*. In addition, we design MABLE, a low-power ad hoc routing protocol for BLE, that works with the BLE's connection-based operation deeply in consideration to improve both reliability and energy efficiency. Lastly, for data transmission, although it seems difficult to make an optimal decision (minimize J/bit) while considering various factors (e.g., data rate, power consumption, and hop distance) of the two different radios, we found out that the solution to the problem is surprisingly simple: use BLE as long as it supports an application's throughput requirement, and use Wi-Fi otherwise. It is worth noting that when *Wi-BLE* chooses Wi-Fi for data transmission, Wi-Fi's high data rate compensates for its high power consumption, resulting in better energy efficiency compared to selecting BLE.

**Contributions:** The contributions of this paper are four-fold.

- We propose *Wi-BLE*, an orchestration layer between IPv6 and link layers for collaborative operation of Wi-Fi and BLE under IPv6. *Wi-BLE* tackles routing and data transmission issues in low-power multihop IoT networks to maximize energy efficiency while satisfying application requirements.
- *Wi-BLE* provides two important findings: (1) While using BLE for routing causes low control overhead, an *opportunistic* use of Wi-Fi for route optimization results in shorter routes with modest increase in control overhead. (2) When an application requires high throughput, using Wi-Fi is better than using BLE in terms of throughput and energy efficiency, making Wi-Fi a reasonable option for *low-power* networks.
- We deign MABLE, an assisting module for AODV that is tightly coupled with BLE's connection-based operation to form reliable routes with low-power consumption. To this end, MABLE features (1) full use of BLE data channels, (2) a newly designed link quality metric, and (3) low-power route recovery.
- We implement MABLE and *Wi-BLE* on Linux and perform performance evaluation on a testbed to show that MABLE operates energy-efficiently and reliably, while
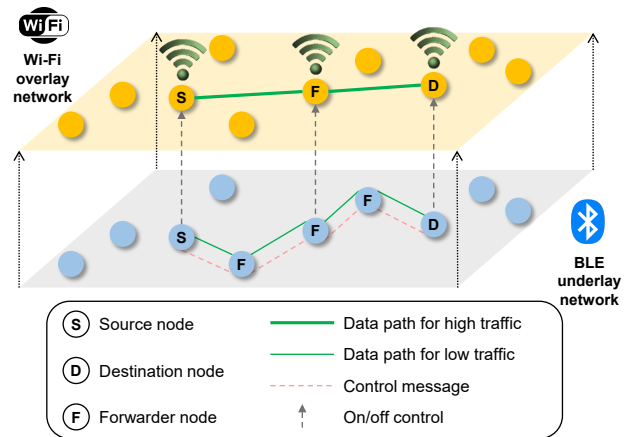
---

[1]This work is an extended version of [3].



Fig. 1.  The concept of Wi-BLE.

*Wi-BLE* flexibly supports high rate services and low energy services. Furthermore, we consider three options for *Wi-BLE* routing, and show which option is proper for a certain communication scenario.

The rest of the paper is organized as follows. In Section 2, we summarize related work. Section 3 describes the overview of *Wi-BLE* system. Section 3 describes the components of MABLE that is a sub-system of *Wi-BLE* , and Section 4 describes *Wi-BLE* operation. In Section 5 we consider detailed environments, work for *Wi-BLE* implementation, and present performance evaluation. Section 6 provides the concluding remarks.

## II. RELATED WORK

### A. Multihop Connectivity for Wi-Fi or BLE

There have been a number of techniques that try to form low-power multihop networks by using either Wi-Fi or BLE. As for Wi-Fi, Independent Basic Service Set (IBSS) and Power Saving Mode (PSM) have been developed for Wi-Fi's multihop connectivity and low-power operation, respectively. Since PSM limits communication opportunities and degrades Wi-Fi's latency or throughput performance, a number of studies have investigated the issues in the context of ad hoc networks [4]–[8]. Most of the studies, however, are theoretical and complex without implementation and evaluation on real devices [4]. In addition, although PSM saves energy, it still requires each Wi-Fi device to periodically wake up and send/receive beacons (control signal) even when there is no data to deliver. In contrast, *Wi-BLE* significantly reduces control signal overhead of Wi-Fi by mainly using BLE (a low-power radio) for control purpose and utilizing Wi-Fi opportunistically for route optimization and high-rate data transmission.

On the other hand, a number of studies, both in academia and industry, have tried to provide multihop connectivity for BLE, which are classified into two types according to the packet relaying method: flooding and routing. BLEmesh [2] is a popular flooding-based scheme standardized by Bluetooth

SIG. Given that a device consumes a significant amount of energy when participating in data packet flooding, BLEmesh enables energy-hungry devices to sleep without participating in flooding and periodically wake up to receive packets from dedicated relay nodes called *friend* nodes. In [9], the authors improve energy efficiency and reliability of flooding by using trickle and gossip algorithms. In [10], the authors reduce the number of transmissions by prioritizing relaying nodes. Despite the efforts, the flooding-based approaches have a fundamental limitation that BLE floods data packets through advertising channels. Unlike data channels, the advertising channels do not support various useful features, such as link-level retransmission, synchronization, resource scheduling, expanded packet length, and extensive channel hopping, which degrades performance. Although concurrent transmission with flooding can be applied to BLE [11], it requires complex time synchronization and violates the BLE standard (lack of compatibility with commercial devices).

As alternatives, there are a number of routing-based schemes that utilize BLE data channels for data transmission. The first attempt in [12] delivers packets through data channels and builds multihop routes by using a simple address-based static routing. As a more advanced approach, ALBER [13] adopts IPv6 routing protocol for low-power and lossy network (RPL) for BLE. ALBER sends routing control packets in advertising channels to discover unconnected neighbors while sending data packets in data channels. It tries to set appropriate parameters for energy-efficient operation in advertising channels and defines a new routing metric considering link quality in data channels. However, given that RPL is a proactive routing protocol, all nodes in a network should participate in route maintenance regardless of the existence of data traffic. In other words, even without data traffic for a long time, each BLE device should consume energy to stay connected to all its neighbors.

For a more general ad-hoc network, a reactive routing protocol is proposed at the application layer [14], which uses hop count as the routing metric and ignores link quality for routing. The work in [15], [16] adopts AODV for BLE and uses RSSI for the routing metric to reflect link quality. However, although AODV is an on-demand routing protocol, the approach in [15] makes all BLE neighbors always keep connected to each other, which is not on-demand from the perspective of BLE link layer. The authors in [16] enable on-demand BLE connection by sending control packets (RREQ and Hello) over advertising channels. An RREQ sender does not broadcast an RREQ but sends it to a single neighbor that has the best RSSI, which causes inefficient routes. Importantly, the two approaches are not implemented on real devices, which limits their practicality. In addition, they do not systematically investigate why RSSI can be a reasonable routing metric in BLE. In contrast, we implement *Wi-BLE* on real devices and investigate how to utilize RSSI reasonably.

Proactive routing schemes such as destination-sequenced distance vector routing (DSDV) [17] and optimized link state routing (OLSR) [18] are not suitable for connection-based BLE networks because they require that a node establishes a connection with every node. Well-known reactive routing

| Platform | Chip | BLE | Wi-Fi |
|---|---|---|---|
| Arduino Primo | nRF52832 | O | O |
| Arduino Vidor | NINA-W10 | O | O |
| RaspbarryPi 3B | BCM2837B0 | O | O |
| RaspberryPi ZeroWH | BCM2835 | O | O |
| Redbear Duo | BCM43438 | O | O |
| Samsung Galaxy S20 | BCM4375 | O | O |
| Aplix MyBeacon | nRF52832 | O | O |
| Minew I3 | nRF52832 | O | O |

TABLE I
MULTI-RADIO CHIPS AND PLATFORMS.

like dynamic source routing (DSR) [19] is also not suitable for BLE becasue of its short frame length. Another reactive routing method, AODV, also has weaknesses such as RREQ flooding, but simple and efficient. Therefore, *Wi-BLE* chooses AODV as the most proper peer-to-peer routing for multi-radio networks and suggests several ways to complement it.

### B. Multi-radio Operation

As shown in Table I, given that many devices support multiple communication interfaces and chip-level multi-radio integration has become common [20], [21], a number of studies have investigated how to improve performance by using multiple different radios collaboratively. A representative way is using a low-power radio as a wake-up radio (WuR): while a high-power radio slepdf most of the time, a low-power radio is always on to monitor the environment and wakes up the high-power radio when necessary (e.g., in the presence of data traffic). In [22], the authors propose a WuR scheme to wake up a high-power radio and show that it outperforms high-power radio-only duty-cycling MACs in terms of energy efficiency, latency, and reliability.

In [23], [24], WuRs are used to improve latency and energy efficiency of LoRa. A downside of vanilla WuR is that it utilizes simple modulation (on-off keying (OOK)) and thus wakes up all the neighbors unnecessarily. The authors in [25] address the problem by using neighbor-specific signal manipulation. The authors in [26] utilize a WuR to collect useful information for the target radio, such as channel availability (i.e., busy or idle). The IEEE 802.11 Working Group recently adopted WuRs in the IEEE 802.11ba standard [27]. Despite its potentials, the WuR approach still has limitations in that it requires custom-designed radios and wake-up signals are vulnerable to security attacks [28].

Some work has investigated the use of ZigBee or BLE with Wi-Fi together. Wi-Zi [29] and Zi-Fi [30] utilize ZigBee to detect and/or send Wi-Fi control packets for scanning and connection establishment, which reduces energy consumption during network initialization. ZPSM [31] sets the wake-up interval of Wi-Fi power saving mode (PSM) to a very large value while using ZigBee as a WuR for on-demand wake-up of Wi-Fi. The Bluetooth core specification has a high speed mode [32] that enables to send Bluetooth packets using Wi-Fi. However, the high speed mode gives up exploiting the feature of Wi-Fi's long transmission range and does not have a clear advantage over Wi-Fi Direct, not implemented on most off-the-shelf BLE devices. ARTPoS [33] determines when to use
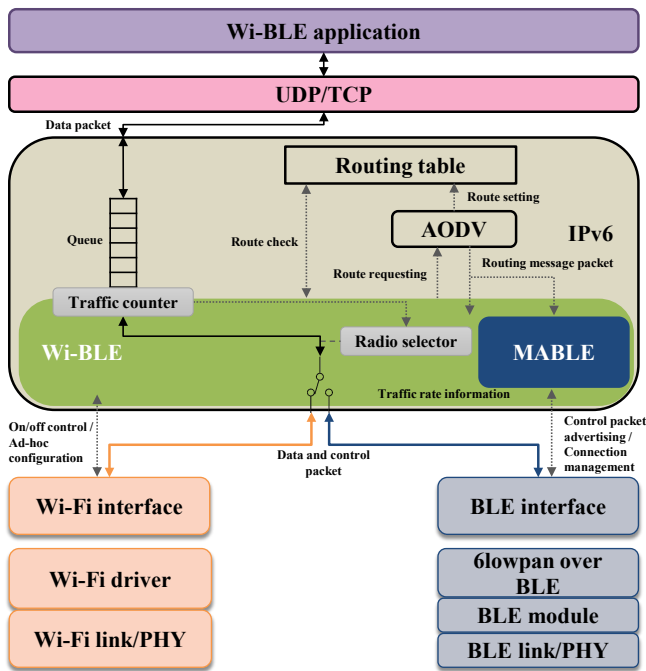
Fig. 2. The protocol stack of Wi-BLE and MABLE.



Fig. 3. An example of traffic flow.

BLE, ZigBee, or Wi-Fi for minimizing transmission power while satisfying a given reliability constraint. However, this scheme does not consider data-rate differences between the three radios, not fully utilizing Wi-Fi's high data rate and large packet size.

While all the works mentioned above do not consider multihop scenarios, dualWireless [34] tries to use ZigBee and Wi-Fi together for improving multihop network performance. Specifically, dualWireless utilizes ZigBee for tree routing and Wi-Fi for data transmission, which achieves both low control overhead and high throughput. However, its tree routing is too simple to be applied in practice. In addition, by giving each radio a dedicated role, it gives up using the capabilities of Wi-Fi's long transmission range for routing and ZigBee's low-power communication for data exchange. In contrast, *Wi-BLE* utilizes BLE and Wi-Fi flexibly both for mesh routing and data transmission to maximize their synergy in multihop ad hoc networks. 6TiSCH++ [35] presents a smart MAC combining BLE and ZigBee. However, it has limitations in concurrent transmission due to violation of standards. Also, this work only can be adopted with two radios that share similar modulation characteristics, it is different from *Wi-BLE*, which considers a combination of Wi-Fi and BLE that have very different characteristics.

*Wi-BLE* is more comprehensive than the prior work in that (1) it considers both routing and data transmission and (2) it minimizes energy consumption while providing required throughput.

## III. SYSTEM OVERVIEW

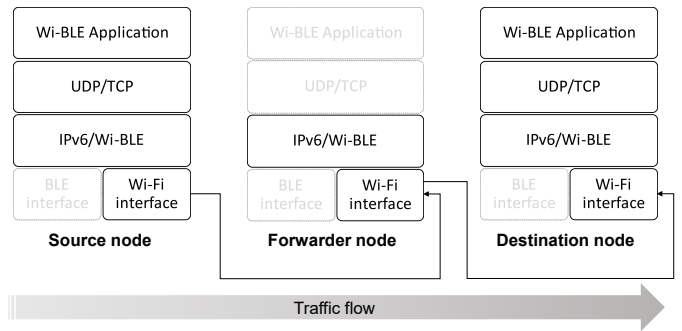Fig. 2 shows the protocol stack of *Wi-BLE*. *Wi-BLE* is a submodule of IPv6 that mediates between two IPv6 operations (i.e., AODV routing and data transmission) and two link layer protocols (i.e., Wi-Fi and BLE). As an orchestration layer between IPv6 and link layer, *Wi-BLE* uses three IPv6 address prefixes to distinguish *Wi-BLE*, BLE, and Wi-Fi.

### A. Control Plane

In the control plane, *Wi-BLE* has **MABLE** that facilitates AODV routing over BLE. AODV control packets (i.e., HELLO, RREQ, RREP, and RERR) are sent/received through BLE by default and BLE connections should be managed together with routes. To this end, MABLE does the following:

- **Channel allocation**: MABLE sets BLE channels to use when sending each routing control packet.
- **Connection management**: MABLE establishes or removes connections between all neighbor nodes in a multihop AODV route.
- **Neighbor table**: MABLE constructs a table that holds each neighbor's link quality information called RSSI.
- **Recovery assist**: MABLE reports connection information and gives a signal about the direction of RERR transmission.

In addition to MABLE for BLE-based AODV routing, we investigate how to efficiently construct multihop routes for Wi-Fi, resulting in the three different routing modes as follows:

- **Mode 1**: Wi-Fi-based AODV routing while using BLE as a wake-up radio
- **Mode 2**: Reusing BLE-based routes for Wi-Fi
- **Mode 3**: Optimization of given BLE-based routes by using Wi-Fi-based AODV routing

### B. Data Plane

In the data plane, *Wi-BLE* has two key components: **traffic counter** and **radio selector**. When a source node generates IPv6 traffic destined for a *Wi-BLE* node, its IPv6 layer sends packets to the *Wi-BLE* module. *Wi-BLE* requests AODV to setup a BLE-based route toward the destination node and while waiting for AODV to construct the route, *Wi-BLE*'s traffic counter measures the application traffic load: how much traffic *Wi-BLE* receives from the application. Once AODV provides a proper route, *Wi-BLE*'s radio selector uses the load information to determine which radio (i.e., BLE or Wi-Fi) to use for data transmission.

When BLE radio is selected, the existing BLE-based route is used for data transmission. When Wi-Fi radio is selected, the source node reuses the BLE-based route or triggers further route optimization for Wi-Fi according to *Wi-BLE* routing mode 3. Importantly, since only the source node can measure its application traffic load toward destination, all forwarders on the multihop route use the same radio that the source selects. Fig. 3 shows an example of the traffic flow from a source node to a destination node when *Wi-BLE* selects Wi-Fi radio for data transmission.

### C. Overall Procedure

*Wi-BLE*'s operation procedure consists of four stages: i) Idle, ii) traffic generation, iii) radio selection, and iv) data transmission.

**Idle stage.** In this stage, MABLE collects neighbor information using BLE. Each node turns on its BLE interface and advertises beacon periodically. When not advertising, each node scans beacons of neighbor nodes and records their RSSI-based link quality.

**Traffic generation stage.** In this stage, the application layer generates traffic where the IPv6 destination address contains *Wi-BLE* interface information. *Wi-BLE* requests MABLE to create a route to the destination node or flood wake-up messages over the network depending on *Wi-BLE* option mode. While MABLE completes the request, *Wi-BLE* measures the traffic generation rate of the application by counting packets accumulated in the queue. When MABLE finishes route generation or flooding, it moves on to the radio selection stage. Note that in *Wi-BLE*, high-power Wi-Fi radio does not involve at all in the control plane operation to save energy .

**Radio selection stage.** In this stage, *Wi-BLE* determines whether Wi-Fi or BLE is suitable for data traffic transmission in terms of energy consumption and throughput based on the measured traffic generation rate of the application. If the traffic generation rate is less than a certain threshold, *Wi-BLE* selects BLE radio for energy efficiency. Otherwise, *Wi-BLE* chooses Wi-Fi radios for high throughput. If Wi-Fi is selected, additional Wi-Fi routing may be performed according to the *Wi-BLE* mode. *Wi-BLE* records the destination and radio interface pair. Pairs of MAC and IP addresses are automatically recorded in the L2 ARP table during Wi-Fi and BLE routing operation. At this time, the address of *Wi-BLE* is recorded, not each radio interface.

**Data transmission stage.** In this stage, *Wi-BLE* sends packets down to the selected radio interface with forwarder IP address referring to the routing table. Each forwarder transmits packets to the next-hop node based on the next-hop IP of the routing table and the ARP table of the indicated radio interface.

## IV. MABLE: AODV ROUTING OVER BLE

In this section, before introducing *Wi-BLE*, we describe MABLE, which performs the peer-to-peer BLE multihop routing function as a sub-module of *Wi-BLE* .
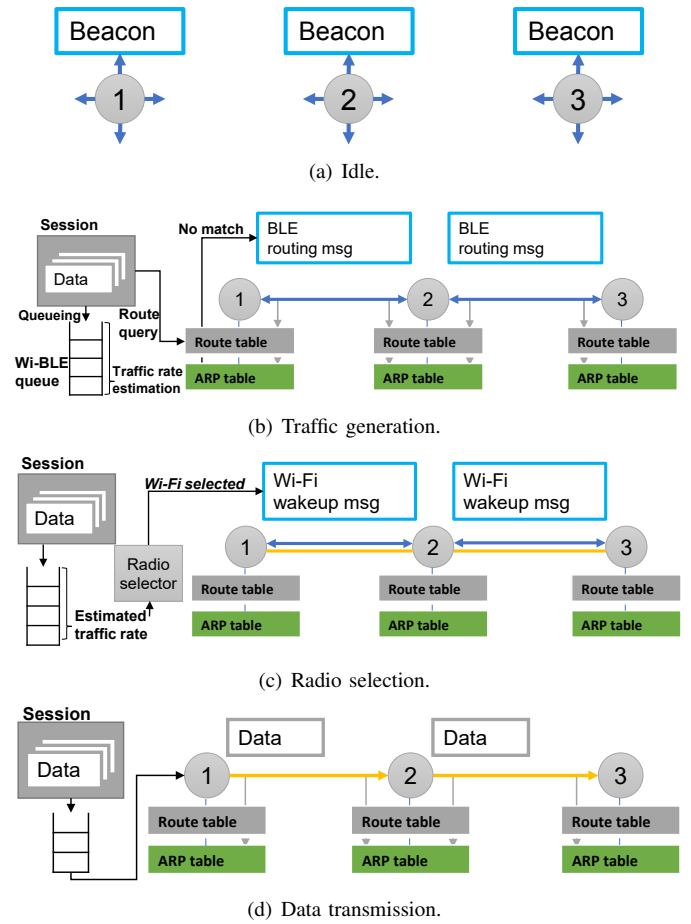


(a) Idle.

(b) Traffic generation.

(c) Radio selection.

(d) Data transmission.

Fig. 4. Operation procedure of Wi-BLE.

### A. BLE Channel Utilization

BLE has two types of channels with different characteristics: 3 *advertising* channels and 37 *data* channels. In advertising channels (connection-less channels), a sender and a receiver are called an advertiser and a scanner, respectively. To enable packet transmissions in a connection-less manner, the advertiser transmits packets through the three advertising channels every advertising interval. The scanner selects an advertising channel every *scan interval*, and scans the channel for a time period called the *scan window*. In advertising channels, the payload length is relatively short, up to 31 bytes, and there is no link-layer ACK. Therefore, packet transmission in advertising channels is limited in terms of throughput and reliability, and thus these channels are mainly used for simple messages, such as beacons[2].

On the other hand, data channels provide connection-based packet transmission between two nodes. If the two nodes establish a connection, they are time-synchronized and share a periodic wake-up schedule and channel hopping sequence. A node called the master informs the other node, called the slave, of information to maintain the connection. Although data channels require control overhead, such as periodic wake-up to maintain time synchronization, these channels support

---

[2]Although there are changes in the latest version of BLE, we describe it based on the baseline for backward compatibility.
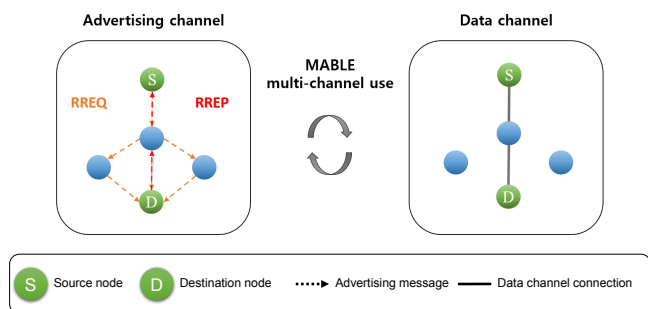
Fig. 5.  Channel multiplexing of MABLE.



Fig. 6.  End-to-end establishment of an AODV route and BLE connections.

payload lengths of up to 251 bytes, achieving better throughput than advertising channels. In addition, packet transmission on data channels is more reliable than that on advertising channels due to link-layer ACK and channel diversity (channel hopping over 37 channels).

Fig. 5 shows how MABLE and *Wi-BLE* utilize the two types of BLE channels. When *Wi-BLE* sends data packets using BLE, it utilizes data channels for high throughput and reliability.[3] However, it is inefficient to allow each BLE node to utilize separate link-layer neighbor discovery and maintain connections with all of its neighbors even when there is no data to send. Thus, MABLE assumes that BLE connections between neighbor nodes do not exist when AODV establishes a route for a source-destination pair, sending routing control packets in advertising channels.

In addition, we found out that RREQ and RREP in IPv6 require payload lengths of 48 bytes and 44 bytes, respectively, which are longer than the maximum length of an advertising packet in BLE. To resolve this issue, we compress AODV control packets by changing their IPv6 addresses to the corresponding BLE addresses (using the address mapping table). This simple compression enables AODV control packet delivery on the BLE advertising channels.

### B. Joint Establishment of Route and Connection

To utilize BLE data channels for application traffic delivery, MABLE should not only support AODV for route construction but also establish all the BLE connections between neighbor nodes on the route. MABLE fulfills the requirements as follows.

**RREQ flooding:** In AODV routing, when a source node establishes a session with a destination node but does not have a route toward the destination, it triggers an RREQ flooding to discover potential forwarders and inform the destination node of the route request. Between the two channel types, MABLE utilizes advertising channels for RREQ flooding, as shown in Fig. 5. As mentioned in Section IV-A, maintaining unnecessarily many BLE connections wastes energy inefficiently. In addition, it is natural to flood short route messages in a best-effort manner and thus, using data channels is overkill.

[3]In addition, 6LoWPAN for BLE mandates that BLE should use data channels to deliver IPv6 packets.
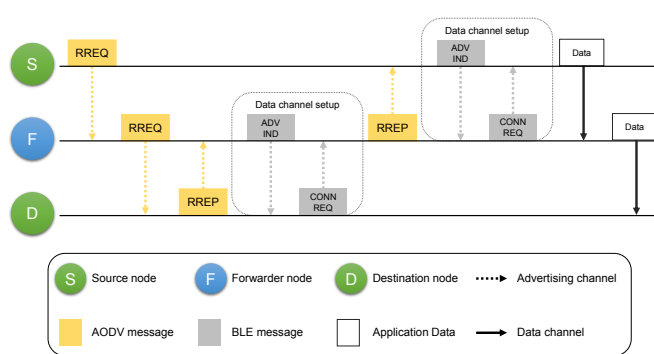
**RREP unicast and Connection establishment:** In AODV, when the destination node receives an RREQ flooded from the source node, it sends (unicasts) an RREP back to the source node via relay nodes which participated in RREQ flooding. Then a bi-directional route between the source and destination nodes is established, which consists of the nodes participating in RREP forwarding. Once the route is set up, *Wi-BLE* utilizes data channels for BLE-based end-to-end data transmission from the source to destination. To this end, MABLE takes care of how to (1) unicast RREP on BLE channels and (2) establish BLE connections between RREP forwarders. The main design choice is when to establish a BLE connection between an RREP sender and receiver, before or after RREP delivery. This is directly related to what type of BLE channels to use for RREP delivery, advertising or data channels. To send an RREP on data channels, a connection between an RREP sender and receiver should be established before sending RREP messages. To this end, each RREQ forwarder (potential RREP receiver in the future), right after sending an RREQ, should send advertising indication on advertising channels for a while, which indicates to RREQ receivers (potential RREP senders in the future) that it is open to connection establishment. However, at the time of sending an RREQ, it is yet undetermined if the RREQ forwarder will be selected as an RREP receiver in the future, meaning that sending advertising indication can end up with energy wastage and unnecessary channel congestion.

To avoid the problem, in MABLE, RREP messages are sent on advertising channels even though they are unicasted. In other words, a BLE connection between an RREP sender and receiver is established after successful RREP delivery. Specifically, as shown in Fig. 6, after sending an RREQ message, the RREQ forwarder (potential RREP receiver) starts scanning advertising channels rather than sending advertising indication, which enables it to receive RREP on advertising channels. After a node receives an RREQ and sends an RREP to a receiver on advertising channels, the RREP sender starts scanning on advertising channels. Once the receiver (previous RREQ forwarder) gets the RREP message while scanning advertising channels, it sends advertising indication on advertising channels to inform the RREP sender of the RREP message's successful delivery (i.e., an implicit ACK).

After the RREP sender receives the advertising indication and sends a connection request, a BLE connection between the two nodes is established, which enables application traffic delivery on data channels in the future.

### C. Link Quality Metric for BLE Data Channels

To establish a stable multihop route, it is important to manage a neighbor table that indicates whether each neighbor node's wireless link quality is good enough. Specifically, in AODV routing, the link quality information is passed through the RREQ flooding procedures. When a node receives an RREQ message from a neighbor node that has bad link quality, it simply ignores the received RREQ, which not only mitigates channel congestion but also improves end-to-end path quality. When AODV operates on BLE, however, the fact that a node recently receives a RREQ message from a neighbor does not necessarily mean that the wireless link between the two nodes is good enough. Given that there are 40 different channels on BLE, the RREQ message may fortunately be sent over one of the best-quality channels.

There are two unique challenges to obtaining a reasonable link quality metric for AODV operation in *Wi-BLE*. First, after a route is set up between a source and a destination, *Wi-BLE* uses data channels for application traffic delivery, meaning that *Wi-BLE* needs good link quality on data channels. While AODV is establishing a route, however, it utilizes advertising channels for control packet delivery. If the link quality of advertising channels is different from that of data channels, AODV routes can be unstable and may break soon.

Second, BLE link layer (called controller part) does not have an interface that provides its detailed information for the upper layer (called host part). For example, when a BLE device sends a packet, the host part does not know how many times the packet was retransmitted, which makes obtaining the famous ETX (Expected Transmission Count) metric unfeasible. In addition, when a BLE device receives a packet, the host part can know the packet's received signal strength indicator (RSSI) but not the specific BLE channel that was used for the packet reception. This means, RSSI values at a single node vary significantly since there are 40 different BLE channels.

To resolve the issues, MABLE exploits the RSSI of AODV control packets (sent on advertising channels) in a particular way. We first perform a preliminary study to investigate how RSSI on advertising channels can represent RSSI on data channels. Specifically, to obtain the RSSI of each BLE channel manually, we establish a connection between two BLE nodes (static) and uses a deterministic hopping sequence that increases the channel number by one. Due to the deterministic hopping, RSSI measured at the slave node can be matched to a specific data channel used for packet transmission. For comparison, we also measure RSSI on advertising channels before establishing the connection. Note that the manual setting is for this specific link measurement study, not practical for real world use cases.

Fig. 7 represents RSSI values on each BLE channel. We consider three scenarios: a good channel scenario where all



(a) Good channel.



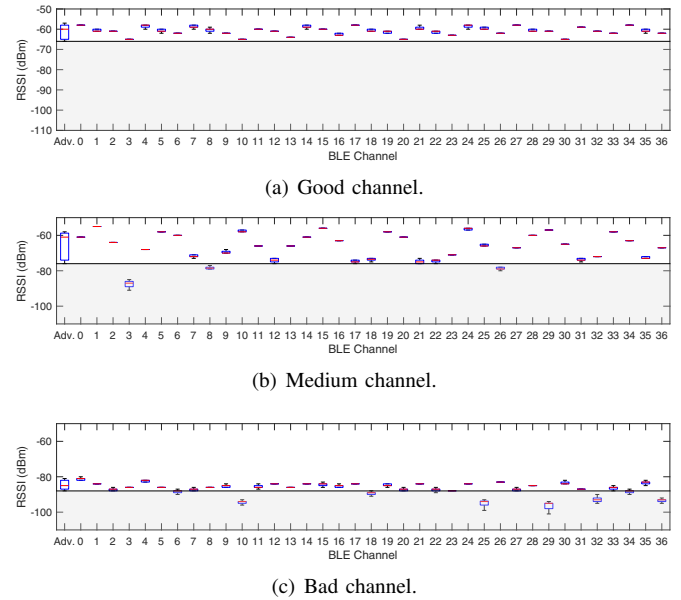(b) Medium channel.



(c) Bad channel.

Fig. 7. Boxplot of RSSIs on BLE channels, showing that the lowest RSSI value on advertising channels can represent the worst-case link quality of all data channels.

packets are successfully delivered, a medium channel scenario where PRR is 80% to 95% due to packet drop in several data channels, and a bad channel scenario where PRR is less than 80% due to disconnection in several channels. In the figures, the regions where RSSI values are lower than the minimum RSSI on advertising channels are marked gray. The experimental results show that in all the three scenarios, most or all data channels are in the white area, meaning that these channels provide RSSI values higher than the lowest RSSI value on advertising channels. In the bad channel scenario, there are relatively more data channels in the gray area (i.e., very bad channels), which are not likely to be used because BLE's adaptive frequency hopping mechanism excludes bad channels from the channel hopping sequence.

The results give us an intuition that although BLE has only three advertising channels (much smaller compared to 37 data channels), if the minimum RSSI value on advertising channels is good enough, packet delivery on data channels will be very reliable. Therefore, for each neighbor, MABLE records the minimum RSSI value of routing packets (sent on advertising channels) and regards its link quality valid when the minimum RSSI is above a given threshold. Specifically, to exclude outliers, we average RSSI on each advertising channel first and get the minimum of the three average RSSI values. Since the BLE controller does not report to the host on which advertising channel the packet is being received, MABLE specifies an advertising channel when sending each periodic beacon or RREQ message and includes the channel information in the message.

### D. Bi-directional Route Error Propagation

Once AODV and MABLE construct a bi-directional route for a source-destination pair, it is important to check validity of each wireless link on the route and rebuild the route if
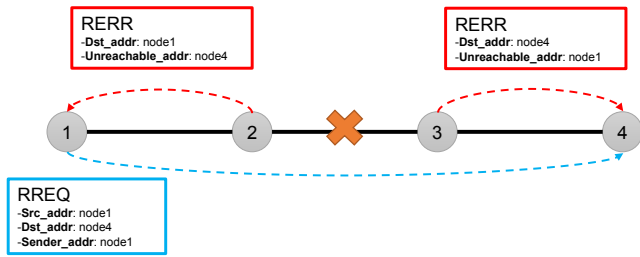
Fig. 8.  Route recovery of MABLE.



(a) One hop.

(b) Multi-hop topology (BLE hop count: 6, Wi-Fi hop count: 2.5).

(c) Multi-hop topology (BLE hop count: 7, Wi-Fi hop count: 2).

Fig. 9.  Energy efficiency of Wi-Fi and BLE.

its quality becomes bad. To this end, AODV nodes check each link status by using HELLO messages or sending data packets through the route. When a node detects that its link toward the next hop node is broken, it sends a Route Error (RERR) message to the source node. All nodes that receive the RERR remove the route information from their routing table and the source node triggers route reconstruction. Although the RERR message is not propagated toward the destination node, the nodes that do not receive RERR also remove the route information when the route lifetime is expired.

**Link status monitoring:** Running AODV over BLE has an advantage in its link status monitoring. Since MABLE establishes a connection for each BLE link on an AODV route and BLE link layer monitors the connection status, MABLE does not need to use a separate L3 method, such as end-to-end packet transmission and HELLO message. Specifically, two BLE nodes of a connected link periodically wake up and exchange null packets even when there is no data to send. If null packets are not exchanged for a given period called the supervision timeout, the controller part of the master node regards the connection as lost and reports this event to its host part. By using the existing BLE function, MABLE detects route failures without additional energy consumption.

**RERR message propagation:** Although a node is deleted from the route table, its BLE connection is still left. In contrast to AODV that sends an RERR to the source node when a link breakage is detected, when MABLE detects a broken BLE connection, it sends an RERR to both source and destination nodes (i.e., bi-directional RERR propagation). This is because MABLE should jointly manage BLE connections and AODV routes. When a route is broken, MABLE needs to disconnect all BLE connections on the route to mitigate redundant energy consumption by operations to maintain connectivity. To this end, when two nodes of a BLE connection experience supervision timeouts, the source-side node and the destination-side node send RERRs to the source and destination, respectively, as shown in Fig. 8. When a node receives an RERR, it removes the corresponding route from its routing table, and if there is no entry in the routing table that has the RERR sender as a next-hop node, it removes the connection with the RERR sender.
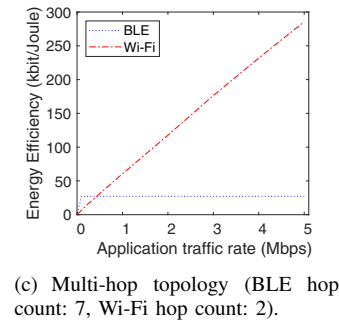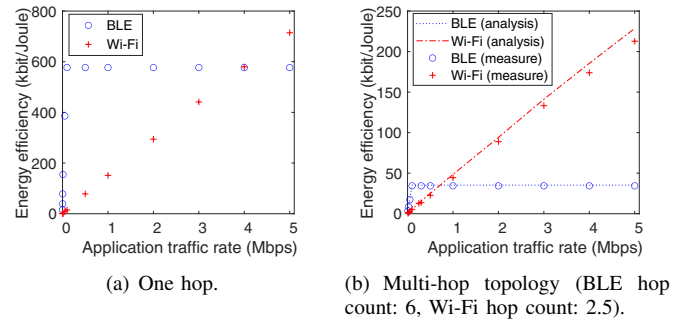
## V.  WI-BLE: WI-FI AD-HOC OVER BLE

Since a source-destination pair has a BLE-based bi-directional route (always-on BLE), this section describes how to operate Wi-Fi synergistically with BLE: (1) when *Wi-BLE* selects Wi-Fi for data transmission instead of BLE, (2) how *Wi-BLE* constructs a multihop route for Wi-Fi, and (3) how *Wi-BLE* wakes up Wi-Fi and puts it to sleep.

### A. Radio Selection

Once a BLE-based route is built for a source-destination pair, the source node selects what radio to use for end-to-end data transmission: Wi-Fi or BLE. We aim to maximize energy efficiency while delivering a given application traffic load. Given that Wi-Fi provides much higher data rate than BLE, using Wi-Fi may be more energy efficient than BLE when the application traffic rate is high. Then, how high should it be? It is a subtle question because application-layer throughput is affected by both the radio data rate and hop distance (route length). We quantitatively investigate this issue by using bit/joule as the performance metric.

Fig. 9 shows energy efficiency of BLE and Wi-Fi according to the application traffic rate in three different scenarios. Fig. 9(a) shows the experimental results in a one-hop (two nodes) scenario. As the application traffic rate increases, energy efficiency of both BLE and Wi-Fi increases. However, the energy efficiency of BLE increases much faster than that of Wi-Fi due to its low-power consumption. In addition, the energy efficiency of BLE saturates much earlier than that of Wi-Fi due to its limited data rate. Finally, the energy efficiencies of the two radios cross when the application traffic rate is 4 Mbps.

(a) Always-on Wi-Fi.

(b) Mode 1: BLE wake-up radio for Wi-Fi.

(c) Mode 2: Reusing BLE routes for Wi-Fi.

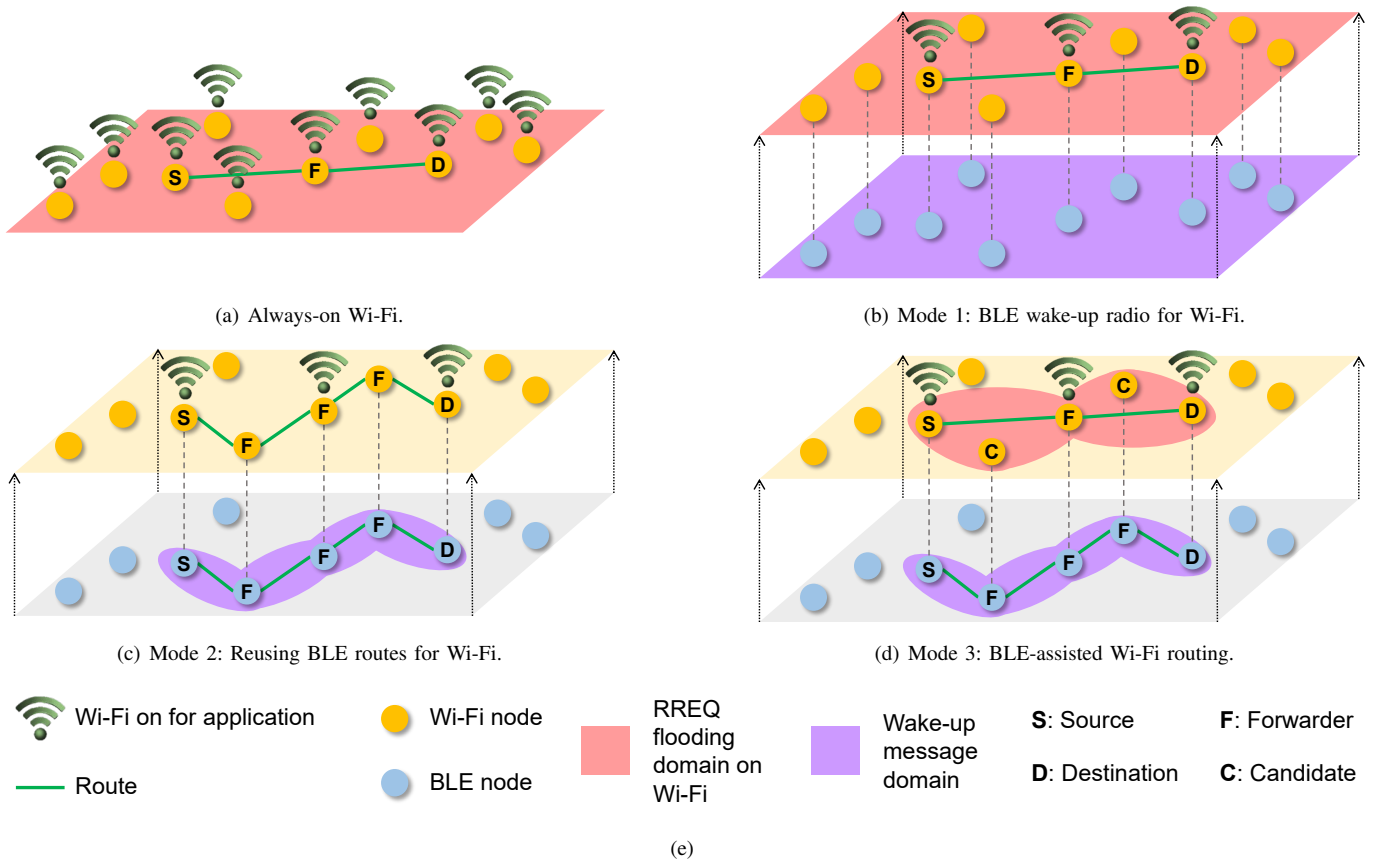(d) Mode 3: BLE-assisted Wi-Fi routing.

(e)

Fig. 10.  Wi-BLE's Wi-Fi routing modes using BLE.

Fig. 9(b) shows the experimental results in a multihop scenario with 6 nodes. Given that Wi-Fi has a longer transmission range, the average number of hops is reduced to 2.5 when using Wi-Fi. The results show that with a longer path, the energy efficiency of BLE is more compromised than that of Wi-Fi. Despite the disadvantage, however, the results show the same pattern as in Fig. 9(a); the energy efficiency of BLE is higher than that of Wi-Fi before it becomes saturated. We also plot analytic values of energy efficiency. These are values obtained by considering topology and end-to-end hop count based on energy efficiency in one-hop. It is also considered that throughput degradation caused by sharing the same collision domain in a Wi-Fi multi-hop situation. Note that the end-to-end throughput in a two-hop network drops by half compared to one-hop on average. As shown in the Fig. 9(b), we can obtain analytic values similar to the experimental values through one-hop energy efficiency and topology information.

Fig. 9(c) considers another scenario that reduces the hop distance more significantly (7 to 2) by using Wi-Fi than the scenario in Fig. 9(b). The larger the hop distance gap between BLE and Wi-Fi, the more BLE will be penalized. The results show, however, BLE's energy efficiency is still higher than Wi-Fi's energy efficiency before BLE gets saturated. Overall, BLE is always more energy efficient than Wi-Fi when the application traffic rate is low enough for BLE to provide.

The results give us a simple solution as follows: (1) A source node measures the application traffic load while an end-to-end BLE route is established. (2) After the route is ready, the source node checks whether the application traffic load is lower than BLE capacity. (3) The source node selects BLE as long as it can cope with the required application traffic rate. Otherwise, it selects Wi-Fi. Although BLE is sometimes more energy efficient than Wi-Fi when the application traffic rate is higher than its capacity, it is impractical to use BLE in this case due to application QoS. Overall, due to the significant energy consumption gap between BLE and Wi-Fi, *Wi-BLE*'s radio selector does not need to consider hop distance or even energy consumption. Instead, a simple comparison between application traffic rate and BLE capacity is sufficient for decision making. We note that other devices would show similar results because they maintain the scale of the difference between Wi-Fi and BLE in energy consumption.

### B. Routing and Radio Wake-up for Wi-Fi

Once the *Wi-BLE* source node decides to use Wi-Fi to deliver application traffic, it needs to build a route toward the destination for Wi-Fi and also manage Wi-Fi's sleep schedule. However, as shown in Fig. 10(a), running AODV separately for Wi-Fi consumes significant energy since many devices turn on Wi-Fi and participate in sending control packets. To minimize Wi-Fi wake-up time during route construction and data transmission, *Wi-BLE* utilizes BLE and its existing routes.

Specifically, *Wi-BLE* provides three routing modes for Wi-Fi as shown in Figs. 10(a), 10(b), and 10(c).

**BLE wake-up radio for Wi-Fi (Mode 1):** Fig. 10(b) depicts Mode 1 operation that utilizes BLE as a wake-up radio for Wi-Fi. In Mode 1, the role of BLE is simple: flooding Wi-Fi wake-up messages on BLE advertising channels over the entire network. Each *Wi-BLE* node that receives a Wi-Fi wake-up message through the BLE radio wakes up its Wi-Fi radio. After a while, the source node starts to construct a Wi-Fi-based multihop route toward the destination by operating AODV. Mode 1 is more energy efficient than using Wi-Fi only since it triggers Wi-Fi routing only when *Wi-BLE* selects Wi-Fi for data transmission. In addition, Mode 1 can construct the best route for Wi-Fi since it investigates all possible routes from scratch. However, all nodes in the network activate Wi-Fi to participate in RREQ flooding even though they might not be part of the final route in the future, which introduces non-trivial routing overhead.

**Reusing BLE routes for Wi-Fi (Mode 2):** Fig. 10(c) depicts Mode 2 operation that simply reuses the existing BLE route for Wi-Fi. Since a route is already prepared for Wi-Fi, *Wi-BLE* only needs to wake up Wi-Fi radios of the nodes on the route. Given that all wireless links on the route have BLE connections, *Wi-BLE* delivers Wi-Fi wake-up messages on BLE data channels, which is more energy efficient than flooding wake-up messages in Mode 1. In addition, Wi-Fi does not operate AODV at all, which significantly reduces control overhead. On the flip side, however, since the existing BLE route is not optimized for Wi-Fi, its hop distance may be longer than expected, resulting in slightly less throughput.

**BLE-assisted Wi-Fi Routing (Mode 3):** Fig. 10(d) depicts Mode 3 operation that optimizes the existing BLE route for Wi-Fi. First, *Wi-BLE* wakes up Wi-Fi radios of the nodes on the BLE route, as in Mode 2. Again, the wake-up message delivery using BLE is energy efficient due to its use of data channels. After a while, the source node starts AODV routing by flooding RREQ as in Mode 1. In contrast to Mode 1, however, only the nodes on the existing BLE route participate in flooding RREQs since other nodes' Wi-Fi radios are still turned off. Utilizing a longer transmission range of Wi-Fi, the AODV routing can find a shorter route than the BLE-based route.

**Wi-Fi wake-up protocol:** To enable BLE-based Wi-Fi wake-up, we define two types of *Wi-BLE* control packets that are delivered using BLE: (1) *Wi-BLE* service REQuest (WREQ) and (2) *Wi-BLE* service REsPonse. The relative roles of WREQ and WREP are similar to those of RREQ and RREP in AODV routing, respectively. They are used for waking up Wi-Fi radios instead of constructing routes. Specifically, an WREQ is flooded (Mode 1) or unicasted along the BLE route (Mode 2 and Mode 3) until it reaches the destination node. Each node that receives an WREQ through its BLE radio turns on its Wi-Fi radio. Once the destination node receives an WREQ message, it generates an WREP message and sends it to the source node as a confirmation that the source node's attempt is successful.

A *Wi-BLE* control packet has four fields: packet type, routing mode, source address, and destination address. The type field contains the control packet type (WREQ or WREP). When *Wi-BLE* receives its control packet through BLE, it checks the type field and starts packet processing. The second field shows the routing mode for Wi-Fi: Mode 1, Mode 2, or Mode 3. The last two fields contain the source and destination addresses of the existing BLE path.

**Wi-Fi turn-off policy:** *Wi-BLE* uses a separate timer to turn off a Wi-Fi radio to reduce unnecessary energy consumption. For example, after a user finishes transmitting streaming data over *Wi-BLE*, the nodes on the Wi-Fi route do not need to turn on their Wi-Fi interfaces. Thus, if the Wi-Fi off timer confirms that there is no packet exchange through the Wi-Fi radio interface for a predetermined time, *Wi-BLE* turns off the Wi-Fi radio and resets the entry of the Wi-Fi route from the routing table.

## VI. EVALUATION

We implement *Wi-BLE* in Linux and evaluate it on real devices. To this end, we configured an indoor testbed as depicted in Fig. 11, where a total of 31 nodes were deployed with a source (node 14) and a destination (node 6). The route snapshot in Fig. 11 shows that *Wi-BLE* provides a 6-hop route by Mode 2, 4-hop route by Mode 3 and Mode 1. For each node, we use a Raspberry Pi device with Broadcom BCM4356 and Atheros AR9271 for BLE and Wi-Fi chipsets, respectively. We use the connection interval of 7.5 msec for BLE and channel 11 with 20 MHz bandwidth for Wi-Fi.

To evaluate the energy efficiency of *Wi-BLE*, we implemented a power monitoring thread in the *Wi-BLE* layer. The thread checks the operation state of Wi-Fi and BLE interfaces at an interval of 1 msec, and measures the power consumption of the communication interfaces according to the power consumption data of each operation state. In addition, it calculates the energy consumed when exchanging packets over each interface according to the packet length and PHY data rate. To do this. we simply modify some code of the BLE module and Wi-Fi driver of the Linux kernel, which allows the feedback route to collect the information of packet exchange between the lower layer and the *Wi-BLE* layer.

### A. BLE Routing

We first evaluate the performance of BLE network with MABLE. Fig. 12 depicts packet reception ratio (PRR) of the end-to-end route between nodes 14 and 6 according to the traffic interval. The source node generates 10 kbps traffic for 10 seconds. The traffic load is low enough to be delivered through BLE once a reliable end-to-end route is constructed. Fig. 12 shows that the PRR of flooding (as in BLEMesh) decreases as the traffic interval decreases due to congestion. In contrast, MABLE shows stable performance regardless of the traffic interval since it utilizes a unicast AODV route, which significantly reduces network congestion. Without our RSSI-based link quality metric, however, MABLE's PRR performance is still limited due to the use of unstable links,
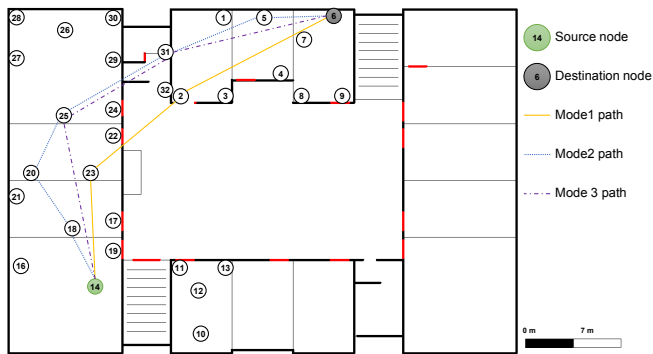
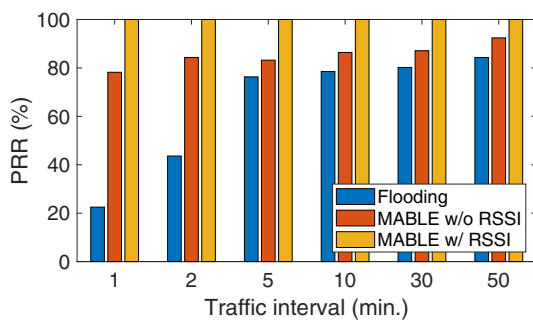Fig. 11. Testbed topology and a route snapshot between nodes 14 and 6.



Fig. 12. Performance of BLE routing schemes.



(a) Reliability.    (b) Energy consumption.

Fig. 13. BLE performance with a link breakage when traffic interval is 5 mins.



(a) End-to-end hop count.    (b) End-toend throughput.

Fig. 14. Traffic delivery performance of Wi-Fi network with varying Wi-Fi transmission power.

resulting in packet loss and route failure. With the RSSI-based link quality metric, MABLE solves both congestion and link quality problems, delivering all packets successfully.

To evaluate the effectiveness of MABLE's route recovery, we forcibly disconnect a wireless link on the route. In that setting, Figs. 13(a) and 13(b) depict packet loss ratio and energy consumption for 5 minutes when the traffic interval is 5 minutes (route recovery is denoted as RR). These figures verify that MABLE's route recovery mechanism further improves its performance. The default AODV detects link breakage due to packet loss while sending traffic over the link, which sacrifices data traffic to detect route errors. In contrast, the route recovery mechanism detects link breakage by using BLE's supervision timeout. It helps to improve packet delivery performance by recovering the route before traffic is sent over a broken link on the route. Also, this shows that BLE, the base link of *Wi-BLE*, responds well to sudden link changes, indicating that *Wi-BLE* operates well under various topologies without performance degradation. In addition, although MABLE reduces energy consumption significantly compared to the flooding approach, the adoption of bi-directional RERR propagation helps to further save energy. This is because BLE connections on unused links are fast removed, avoiding unnecessary null packet exchanges.

### B. Wi-Fi Routing over BLE

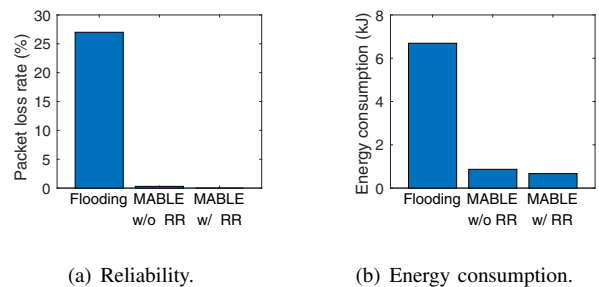We evaluate the performance of the overlay Wi-Fi network according to the three Wi-Fi routing modes of *Wi-BLE*: (1)

BLE wake-up radio for Wi-Fi, (2) Reusing BLE routes for Wi-Fi, and (3) BLE-assisted Wi-Fi routing. Given that *Wi-BLE* selects Wi-Fi for traffic delivery when traffic load is heavy, we generate 6 Mbps UDP traffic. Specifically, the source node generates 6 Mbps traffic for the first half of the given traffic interval and rests for the other half. For comparison, we also evaluate the case of using Wi-Fi only.

Figs. 14(a) and 14(b) show end-to-end hop count and throughput according to the Wi-Fi transmission power when the traffic interval is 30 minutes. While it is obvious that the Wi-Fi-only case provides the best performance in all scenarios, it is important to observe how similar the performance of each routing mode is to its best performance. First of all, except Mode 2 that reuses BLE routes for Wi-Fi, all the schemes experience performance improvement in terms of hop count and throughput as Wi-Fi transmission power increases. Although reusing BLE routes is efficient in that it nullifies Wi-Fi routing overhead, it cannot utilize Wi-Fi's larger coverage, which is a trade-off to consider for practical use.

In addition, Mode 1 shows similar performance compared to the Wi-Fi-only case because it uses Wi-Fi to construct a completely new route while using BLE as a wake-up radio only; it examines all candidates again from scratch. Lastly, in Mode 3, hop count and throughput performances are significantly better than those in Mode 2 and they are comparable to those in Mode 1. Although Mode 3 discovers Wi-Fi routes based on the limited set of nodes on pre-constructed BLE routes, the results show that the impact of this limited discovery is significant.

In practice, traffic delivery performance should be considered together with energy consumption. To this end, we plot the power consumption in each routing mode according to traffic interval in Fig. 15. We used lines in the middle *Wi-BLE*
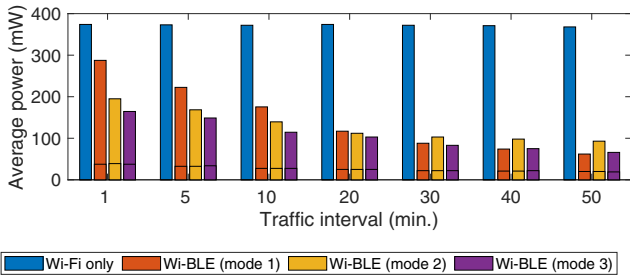
Fig. 15. Average power consumption of Wi-Fi network according to the traffic interval.



(a) Light traffic case (10 kbps).



(b) Heavy traffic case (6 Mbps).

Fig. 16. Energy efficiency with varying traffic interval when *Wi-BLE*'s data radio is manually selected.

performance bars to separate energy consumption between Wi-Fi and BLE. The top and bottom represent the energy consumed by Wi-Fi and BLE interfaces, respectively. As in Fig. 14, the source node generates 6 Mbps UDP traffic for the first half of a given traffic interval. Thus, while the total traffic load is the same in all cases, the traffic becomes more bursty as the traffic interval increases. Since AODV removes routes after an application session ends, a short traffic interval causes more routing overhead due to frequent route construction.

In contrast to those in Figs. 14(a) and 14(b), the results in Fig. 15 shows that using only Wi-Fi provides the worst performance, significantly worse than all the routing modes of *Wi-BLE*. Without a practical sleep mechanism for ad hoc networks, Wi-Fi-only network shows similar performance regardless of the traffic interval since the radios are always on. This demonstrates why Wi-Fi is not preferred over BLE in many cases despite its high throughput performance.

On the other hand, all the three routing modes in *Wi-BLE* only turn on Wi-Fi when necessary, resulting in lower power consumption as traffic interval increases due to less routing overhead. The energy consumed in transmitting control messages in the BLE network is negligible compared to that in transmitting Wi-Fi data. The three modes have a trade-off according to how frequently routes are reconstructed. Specifically, Mode 1 constructs the best route for Wi-Fi by consuming a significant amount of energy, meaning that longer traffic intervals result in relatively better power consumption (i.e., using constructed routes for a long time). Mode 2 has low routing overhead, resulting in lower power consumption compared to Mode 1 when the traffic interval is short. However, due to its inefficient routes, Mode 2 consumes more power to deliver data, showing worse performance than Mode 1 when the traffic interval is long. Lastly, Mode 3, an improved version of Mode 2, always provides better energy efficiency than Mode 2 and thus the best (or nearly the best) performance in all cases.

*C. Radio Selection*

In this section, we evaluate the effectiveness of *Wi-BLE*'s radio selection in terms of energy efficiency (joule per bit), which is a comprehensive metric that includes both throughput and energy consumption. To this end, Figs. 16(a) and 16(b) depict energy efficiency of four *Wi-BLE* options with the radio
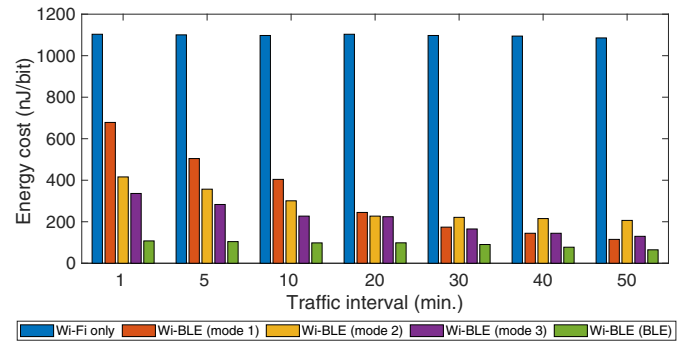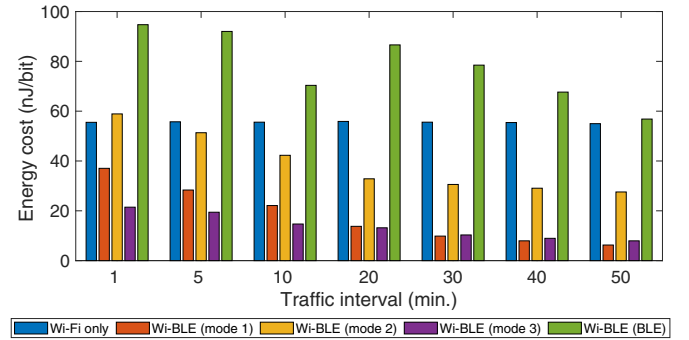
selection disabled. Specifically, Modes 1, 2, and 3 force Wi-Fi to be used while *Wi-BLE* (BLE) should use BLE for traffic delivery.

Fig. 16(a) shows energy efficiency when the traffic rate is 10 kbps. With the light traffic, *Wi-BLE*'s radio selector chooses BLE for data forwarding. The results show that while ranking between the three Wi-Fi operation modes varies according to traffic interval, *Wi-BLE* (BLE) always provides the best energy efficiency. This confirms that using BLE for data delivery when traffic load is low enough is a reasonable choice in terms of energy consumption and data delivery.

Fig. 16(b) shows energy efficiency when the traffic rate is 6 Mbps. Given that the traffic load exceeds the BLE capacity, *Wi-BLE*'s radio selector chooses Wi-Fi for data delivery. In contrast to Fig. 16(a), *Wi-BLE* (BLE) always provides the worst performance in Fig. 16(b). Despite its low-power characteristic, BLE provides too low throughput compared to Wi-Fi, resulting in worse energy efficiency. This confirms that using Wi-Fi in a low-power multihop network is viable when traffic load is high. Mode 3 provides the best (or nearly the best) performance in all cases, verifying the effectiveness of using BLE actively in a Wi-Fi multihop network.

## VII. CONCLUSION

It is probably easy to imagine that multiple communication interfaces with different characteristics can be used together to achieve better performance. However, realizing this with real devices and software stacks is a non-trivial challenge. In this

paper, we have investigated the cooperative use of Wi-Fi and BLE for routing and data forwarding in low-power multihop networks. First of all, to maximize the potential of BLE, we have proposed MABLE that improves ad hoc routing by deeply considering two types of link layer operation in BLE: connection-based operation of data channels and connection-less operation of advertising channels.

Building on MABLE, we have extensively investigated how to utilize existing BLE routes for discovering Wi-Fi routes with less routing overhead. Experimental results have shown the pros and cons of the three routing modes according to how cooperatively use BLE and Wi-Fi, clearly revealing the practical benefits of using BLE on Wi-Fi routes. Lastly, *Wi-BLE* shows that with the help of careful routing and radio wake-up strategies, using Wi-Fi for data transmission can be a better option than using only BLE in low-power multi-hop networks.

## REFERENCES

[1] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Jul. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3561.txt

[2] M. W. Group *et al.*, "Mesh profile—bluetooth specification," 2017.

[3] T. Lee, "Ble connectivity and its multi-hop extension for iot applications," Ph.D. dissertation, Seoul National University, 2017.

[4] V. Vukadinovic, I. Glaropoulos, and S. Mangold, "Enhanced power saving mode for low-latency communication in multi-hop 802.11 networks," *Ad hoc networks*, vol. 23, pp. 18–33, 2014.

[5] A. Belghith, W. Akkari, and J.-M. Bonnin, "Traffic aware power saving protocol in multi-hop mobile ad-hoc networks." *J. Netw.*, vol. 2, no. 4, pp. 1–13, 2007.

[6] A. Belghith and W. Akkari, "Power saving mechanisms for ad hoc networks based on handshaking information tapping," in *Proc. VECoS*, 2008, pp. 1–11.

[7] K. Lee, J. Y. Choi, W. H. Kwon, and H. S. Park, "An energy-efficient contention-based mac protocol for wireless ad hoc networks," in *Proc. IEEE VTC*, 2006, pp. 430–434.

[8] J. Wang, Y. Fang, and D. Wu, "A power-saving multi-radio multi-channel mac protocol for wireless local area networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[9] A. Gogic, A. Mujcic, S. Ibric, and N. Suljanovic, "Performance analysis of bluetooth low energy mesh routing algorithm in case of disaster prediction," *International J. Comput. Inf. Engineering*, vol. 10, no. 6, pp. 1075–1081, 2016.

[10] H.-S. Kim, J. Lee, and J. W. Jang, "Blemesh: A wireless mesh network protocol for bluetooth low energy devices," in *Proc. IEEE FiCloud*, 2015, pp. 558–563.

[11] B. Al Nahas, S. Duquennoy, and O. Landsiedel, "Concurrent transmissions for multi-hop bluetooth 5." in *Proc. EWSN*, 2019, pp. 130–141.

[12] B. K. Maharjan, U. Witkowski, and R. Zandian, "Tree network based on bluetooth 4.0 for wireless sensor network applications," in *Proc. IEEE EDERC*, 2014, pp. 172–176.

[13] T. Lee, M.-S. Lee, H.-S. Kim, and S. Bahk, "A synergistic architecture for rpl over ble," in *Proc. IEEE SECON*, 2016, pp. 1–9.

[14] K. Mikhaylov and J. Tervonen, "Multihop data transfer service for bluetooth low energy," in *Proc. IEEE ITST*, 2013, pp. 319–324.

[15] C. Jung, K. Kim, J. Seo, B. N. Silva, and K. Han, "Topology configuration and multihop routing protocol for bluetooth low energy networks," *IEEE Access*, vol. 5, pp. 9587–9598, 2017.

[16] C. Jung and K. Han, "Maximum power plus rssi based routing protocol for bluetooth low energy ad hoc networks," *Wireless Commun. Mobile Comput.*, vol. 2017, 2017.

[17] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM Comput. Commun. Review*, vol. 24, no. 4, pp. 234–244, 1994.

[18] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol (OLSR)," 2003.

[19] D. B. Johnson, D. A. Maltz, J. Broch *et al.*, "Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks," *Ad hoc Netw.*, vol. 5, no. 1, pp. 139–172, 2001.

[20] *CC1352R SimpleLink™ High-Performance Multi-Band Wireless MCU*, Texas Instruments, 2021.

[21] *Qualcomm® QCA4020/24 Multi-mode SoCs*, Qualcomm, 2019.

[22] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?" *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 674–687, 2015.

[23] F. A. Aoudia, M. Gautier, M. Magno, M. Le Gentil, O. Berder, and L. Benini, "Long-short range communication network leveraging lora™ and wake-up receiver," *Microprocessors Microsyst.*, vol. 56, pp. 184–192, 2018.

[24] F. A. Aoudia, M. Magno, M. Gautier, O. Berder, and L. Benini, "A low latency and energy efficient communication architecture for heterogeneous long-short range communication," in *Proc. IEEE DSD*, 2016, pp. 200–206.

[25] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems," in *Proc. IEEE INFOCOM*, 2015, pp. 522–530.

[26] S. Tang, C. Zhang, H. Yomo, and S. Obana, "Energy and spectrum efficient wireless lan by tightly integrating low-power wake-up radio," in *Proc. IEEE PIMRC*, 2016, pp. 1–6.

[27] D.-J. Deng, M. Gan, Y.-C. Guo, J. Yu, Y.-P. Lin, S.-Y. Lien, and K.-C. Chen, "Ieee 802.11 ba: Low-power wake-up radio for green iot," *IEEE Commun. Mag.*, vol. 57, no. 7, pp. 106–112, 2019.

[28] A. T. Capossele, V. Cervo, C. Petrioli, and D. Spenza, "Counteracting denial-of-sleep attacks in wake-up-radio-based sensing systems," in *Proc. IEEE SECON*, 2016, pp. 1–9.

[29] T. Jin, G. Noubir, and B. Sheng, "Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access," in *Proc. IEEE INFOCOM*, 2011, pp. 1593–1601.

[30] Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, and J. Ma, "Zifi: Exploiting cross-technology interference signatures for wireless lan discovery," *IEEE Trans. Mobile Comput.*, vol. 13, no. 11, pp. 2675–2688, 2014.

[31] H. Qin and W. Zhang, "Zigbee-assisted power saving for more efficient and sustainable ad hoc networks," *IEEE Trans. Wireless Commun*, vol. 12, no. 12, pp. 6180–6193, 2013.

[32] S. Bluetooth, "Bluetooth 3.0," *HS specification*, vol. 21, 2009.

[33] D. Mu, Y. Ge, M. Sha, S. Paul, N. Ravichandra, and S. Chowdhury, "Robust optimal selection of radio type and transmission power for internet of things," *ACM Trans. Sensor Netw.*, vol. 15, no. 4, pp. 1–25, 2019.

[34] I. Foche-Pérez, J. Simó-Reigadas, I. Prieto-Egido, E. Morgado, and A. Martínez-Fernández, "A dual ieee 802.11 and ieee 802.15–4 network architecture for energy-efficient communications with low-demanding applications," *Ad Hoc Netw.*, vol. 37, pp. 337–353, 2016.

[35] M. Baddeley, A. Aijaz, U. Raza, A. Stanoev, Y. Jin, M. Schuß, C. A. Boano, and G. Oikonomou, "6tisch++ with bluetooth 5 and concurrent transmissions," in *Proc. EWSN*, 2021, pp. 25–30.

**Myungsup Lee** received the B.S. degree from SNU, Seoul, Korea in 2014 in Electrical and Computer Engineering. He is currently working towards the Ph.D. degree in Electrical and Computer Engineering at SNU. His research interests include sensor network, Bluetooth low energy, and multi-interface cooperative system.

**Taeseop Lee** received the B.S. degree in Electrical and Electronic Engineering from Seoul National University (SNU), in 2011, and the Ph.D. degree from SNU, in 2017. He is currently a staff engineer in Samsung Research, Seoul. His research interests include artificial intelligence based SON and reinforcement learning based base station operation.

**Hyung-sin Kim** (Member, IEEE) received the B.S. degree in Electrical Engineering and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Science (EECS) from Seoul National University (SNU), Seoul, South Korea, in 2009, 2011, and 2016, respectively, all with outstanding thesis awards. He was a Postdoctoral Scholar at Network Laboratory (NETLAB), SNU, until August 2016 and Real-time, Intelligent, Secure, Explainable systems (RISELab), University of California, Berkeley, until August 2019, and a Software Engineer at Google Nest until February 2020. He received the Qualcomm Fellowship in 2011 and the National Research Foundation (NRF) Global Ph.D. Fellowship and Postdoctoral Fellowship in 2011 and 2016, respectively. He is currently an Assistant Professor at Graduate School of Data Science, SNU. His research interest includes Internet of things and ambient artificial intelligence.

**Saewoong Bahk** (Senior Member, IEEE) received the B.S. and M.S. degrees in Electrical Engineering from Seoul National University (SNU), in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania, in 1991. He was with AT&T Bell Laboratories as a Member of Technical Staff, from 1991 to 1994, where he had worked on network management. From 2009 to 2011, he served as the Director of the Institute of New Media and Communications. He is currently a Professor at SNU. He has been leading many industrial projects on 3G/4G/5G and the IoT connectivity supported by Korean industry. He has published more than 300 technical articles and holds more than 100 patents. He is a member of the National Academy of Engineering of Korea (NAEK). He was a recipient of the KICS Haedong Scholar Award, in 2012. He was President of the Korean Institute of Communications and Information Sciences (KICS). He has been serving as Chief Information Officer (CIO) of SNU. He was General Chair of the IEEE WCNC 2020 (Wireless Communication and Networking Conference), IEEE ICCE 2020 (International Conference on Communications and Electronics), and IEEE DySPAN 2018 (Dynamic Spectrum Access and Networks). He was Director of the Asia–Pacific Region of the IEEE ComSoc. He is an Editor of the IEEE Network Magazine and IEEE Transactions on Vehicular Technology. He was TPC Chair of the IEEE VTC-Spring 2014, and General Chair of JCCI 2015, Co-Editor-in-Chief of the Journal of Communications and Networks (JCN), and on the Editorial Board of Computer Networks Journal and the IEEE Tran. on Wireless Communications.