Ph.D. DISSERTATION

Improving Network Performance through Cooperation Between Heterogeneous IoT Devices

이기종 IoT 기기간 협력을 통한 네트워크 성능 향상

BY

Myungsup LEE

AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Improving Network Performance through Cooperation Between Heterogeneous IoT Devices

이기종 IoT 기기간 협력을 통한 네트워크 성능 향상

BY

Myungsup LEE

AUGUST 2022

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Improving Network Performance through Cooperation Between Heterogeneous IoT Devices

이기종 IoT 기기간 협력을 통한 네트워크 성능 향상

지도교수 박 세 웅

이 논문을 공학박사 학위논문으로 제출함

2022년 8월

서울대학교 대학원

전기 정보 공학부

이 명 섭

이명섭의 공학박사 학위 논문을 인준함

2022년 8월

위 원 장:	최 완	(인)
부위원장:	박 세 웅	(인)
위 원:	이 경 한	(인)
위 원:	김 형 신	(인)
위 원:	백 정 엽	(인)

Abstract

The Internet of Things (IoT) has become a daily life by pioneering applications in various fields. In this dissertation, we consider increasing transmission data rate with energy efficiency, extending transmission coverage with low power, and improving reliability in congested frequency bands as three challenges to expanding IoT applications. We address two issues to overcome these challenges.

First, we design a layered network system with a new structure that combines Bluetooth Low Energy (BLE) and Wi-Fi networks in a multi-hop network. Based on the system, we propose methods to increase data rate with energy efficiency and extend transmission coverage in a low-power situation. We implement the proposed system in the Linux kernel and evaluate the performance through an indoor testbed. As a result, we confirmed that the proposed system supports high data traffic and reduces average power consumption in the testbed compared to the existing single BLE/Wi-Fi ad-hoc network in a multi-hop situation.

Second, we tackle the adaptive frequency hopping (AFH) problem of BLE through cross-technology communication (CTC) and channel weighting. We design the AFH scheme that weights the channels used by BLE devices with improving reliability in the congested bands of both Wi-Fi and BLE devices. We evaluate the proposed scheme through prototype experiments and simulations, confirming that the proposed scheme increases the packet reception rate of BLE in the congested ISM band compared to the existing AFH algorithm.

keywords: Internet of Things(IoT), multi-hop network, Wi-Fi, BLE, frequency hopping

student number: 2014-21743

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

IoT refers to numerous ’things’ connected to the Internet and can share data with other things, such as IoT applications, network-connected devices, or industrial equipment. Internet-connected devices use built-in sensors to collect data and, in some cases, react accordingly. IoT-connected devices and machines help improve the way we work and live. IoT is being applied in various fields, from smart home devices that automatically control heating and lighting to smart factories that monitor industrial equipment to find problems and automatically solve them to prevent failures.

As IoT applications become a daily life and AI technology develops, IoT dreams of more and more new applications, and the number of IoT devices installed around us is only increasing.

Despite these opportunities, wireless communication technologies for IoT are having difficulties adapting to new environments internally and externally.

First, as IoT applications and types of data traffic diversify, it has become difficult to support new applications with radios with low data rates for energy efficiency.

Second, as the range of applications targeted by IoT expands to the number of km, expanding coverage has become a new challenge while operating with sustainable

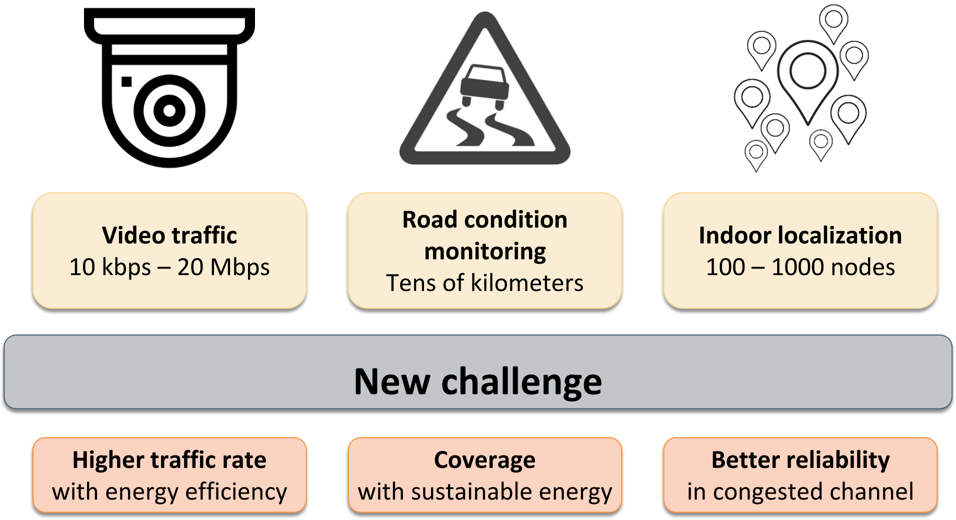


Figure 1.1: New application challenges of IoT.

energy.

Third, as the number of IoT devices becomes very large and the bands shared become congested, it is no longer possible to solve such congestion with existing collision avoidance methods.

In this dissertation, with motivation from the above three issues, we study methods to break through the these limitations of IoT.

1.2 Contributions and Outline

In general, IoT is targeted at battery-limited devices. The current operations of IoT device is mainly designed for low data traffic. However, as applications of IoT are being diversified and IoT devices are becoming massive, the capacity limitation of IoT is the lack of a new IoT world. Can't we reach higher data rates while still maintaining low power? Can IoT devices with limited energy cover a wide range of applications? Will current interference avoidance techniques still work well in more massive IoT environments?

In this dissertation, we investigate these three issues that need to be addressed to overcome IoT's limitation. This dissertation is organized as follows.

In Chapter 2, we proposed a novel layered architecture using Wi-Fi and BLE for higher traffic rate with energy efficiency and coverage with sustainable energy. We design *Wi-BLE* to maintain a network with low power consumption and also provide a high data rate occasionally. For the *Wi-BLE* implementation, we designed MABLE first to maintain the underlay BLE network with low energy consumption. We also design the radio selection module for energy-efficient data transmission with given application data traffic. Finally, we evaluated the performance of *Wi-BLE* through extensive experiments in an indoor testbed. We compared the performance results with that of the Wi-Fi network using the AODV routing protocol. The results show that our proposed *Wi-BLE* significantly reduces the energy consumption with high data traffic in our testbed.

In Chapter 3, we addressed the frequency hopping issues of BLE in the congested ISM band for better reliability in congested channel. We proposed a new adaptive frequency hopping algorithm named *WBC-AFH* exploiting CTC to evaluate each BLE channel with low energy cost. We mathematically derive an optimal size of channel set for a given channel condition for the designed *WBC-AFH* system. *WBC-AFH* adjusts the size of the frequency hopping set in a weighted manner. We evaluate the performance of *WBC-AFH* through extensive simulation and prototype experiments based on implementation. The results show significant performance improvement compared to conventional AFH of various methods. We show significant improvement in the reliability of BLE with *WBC-AFH*.

We conclude the dissertation in Chapter 4.

Chapter 2

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

2.1 Introduction

Wireless 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 chapter 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

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

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 *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 chapter 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.

2.2 Related Work

2.2.1 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

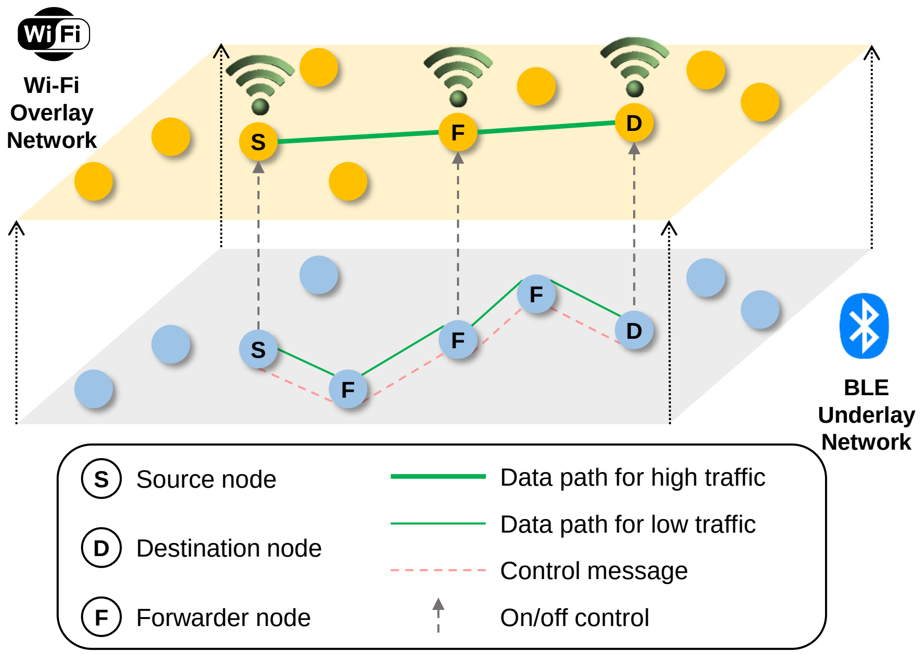


Figure 2.1: The concept of Wi-BLE.

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 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.

2.2.2 Multi-radio Operation

Table 2.1: Multi-radio chips and platforms

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

As shown in Table 2.1, 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 [29]. Wi-Zi [30] and Zi-Fi [31] utilize ZigBee to detect and/or send Wi-Fi control packets for scanning and connection establishment, which reduces energy consumption during network initialization. ZPSM [32] 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 [33] 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 [34] determines when to use 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 [35] 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++ [36] 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.

2.3 System Overview

Figure 2.2 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.

2.3.1 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.

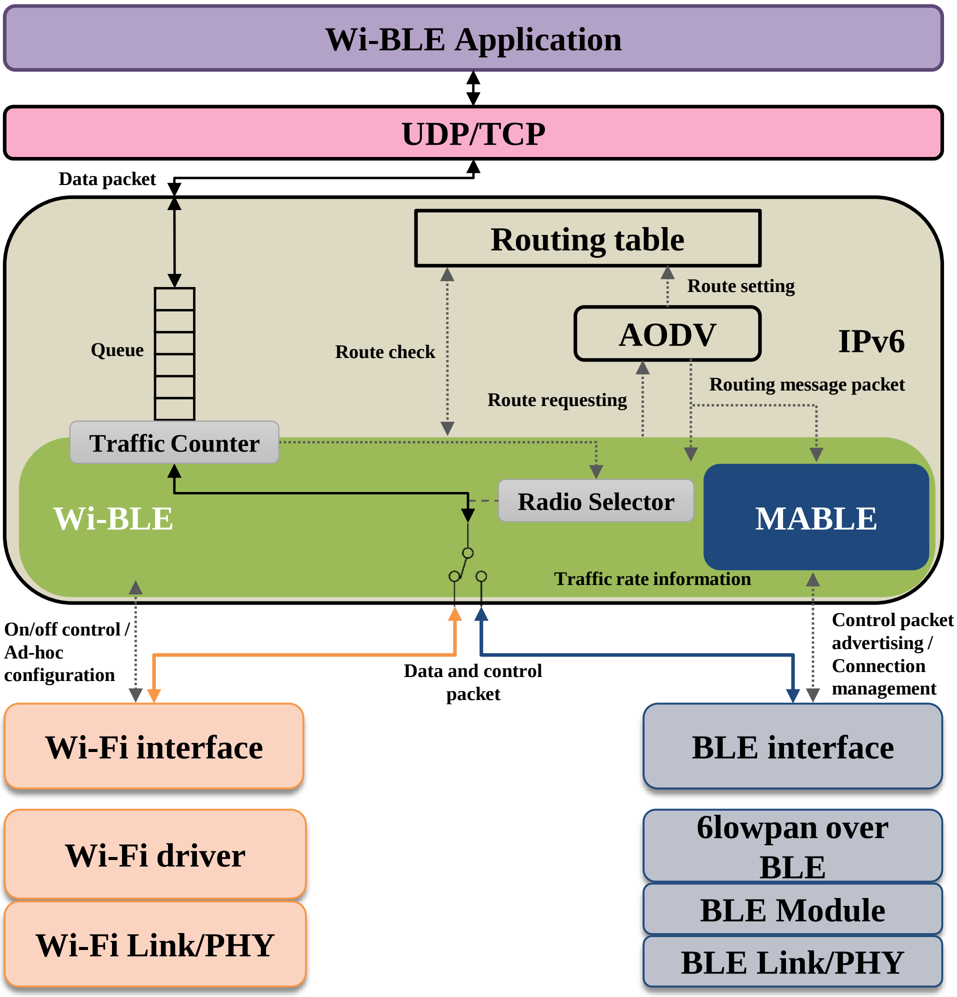


Figure 2.2: The protocol stack of Wi-BLE and MABLE.

- **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

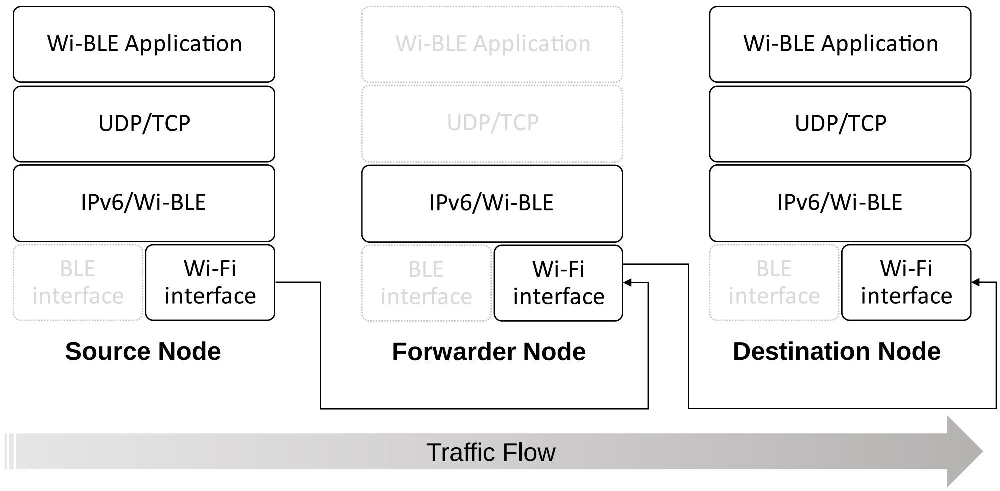


Figure 2.3: An example of traffic flow.

2.3.2 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. Figure 2.3 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.

2.3.3 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.

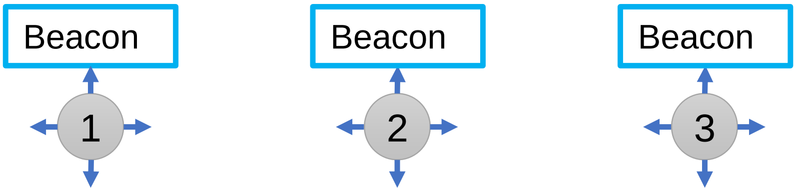
Data transmission stage. In this stage, *Wi-BLE* sends packets down to the selected radio interface with forwarder IP address referring to the routing table. Each forwarder transmits packets to the next-hop node based on the next-hop IP of the routing table and the ARP table of the indicated radio interface.

2.4 MABLE: AODV Routing over BLE

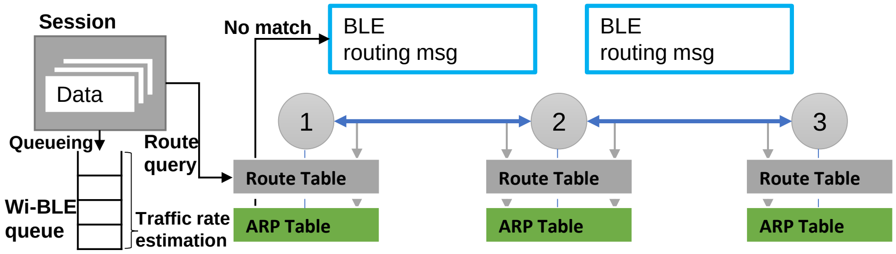
In this section, before introducing *Wi-BLE*, we describe MABLE, which performs the peer-to-peer BLE multihop routing function as a sub-module of *Wi-BLE* .

2.4.1 BLE Channel Utilization

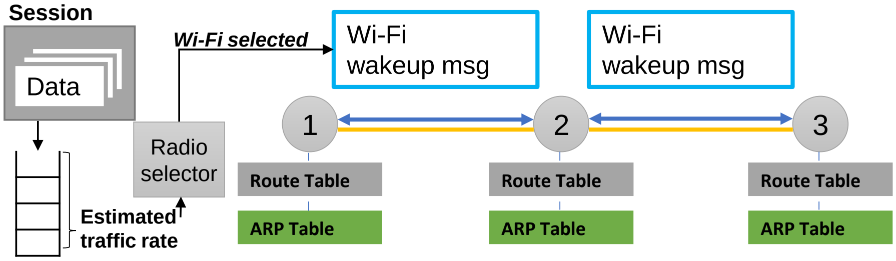
BLE has two types of channels with different characteristics: 3 *advertising* channels and 37 *data* channels. In advertising channels (connection-less channels), a sender and a receiver are called an advertiser and a scanner, respectively. To enable packet transmissions in a connection-less manner, the advertiser transmits packets through



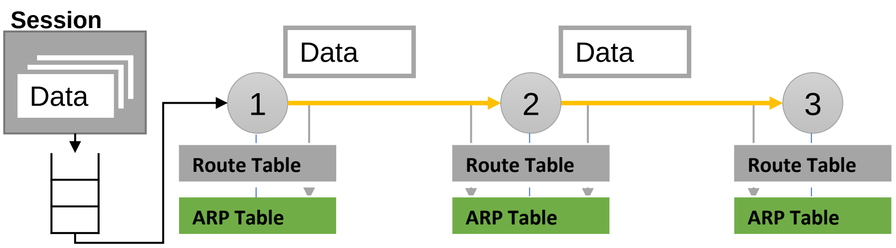
(a) Idle



(b) Traffic generation



(c) Radio selection



(d) Data transmission

Figure 2.4: Operation procedure of Wi-BLE

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].

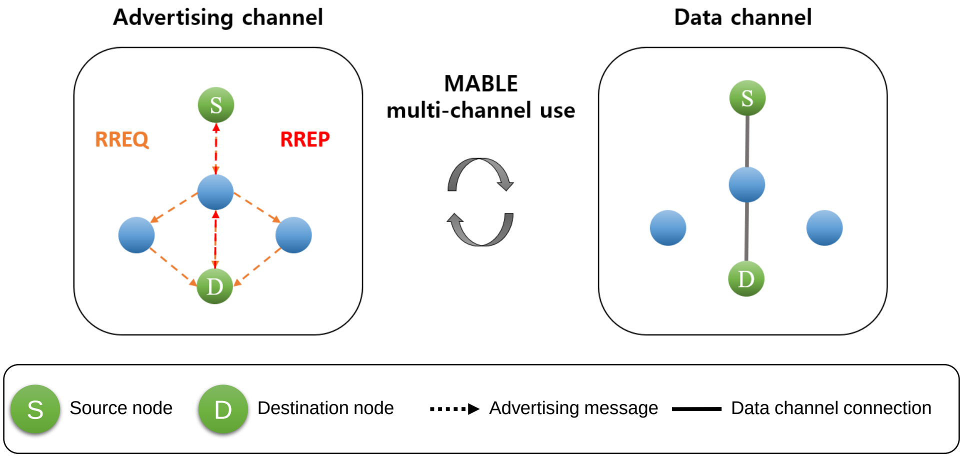


Figure 2.5: Channel multiplexing of MABLE.

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 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

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

link-layer ACK and channel diversity (channel hopping over 37 channels).

Figure 2.5 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.

2.4.2 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 Figure 2.5. As mentioned in Section 2.4.1, 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.

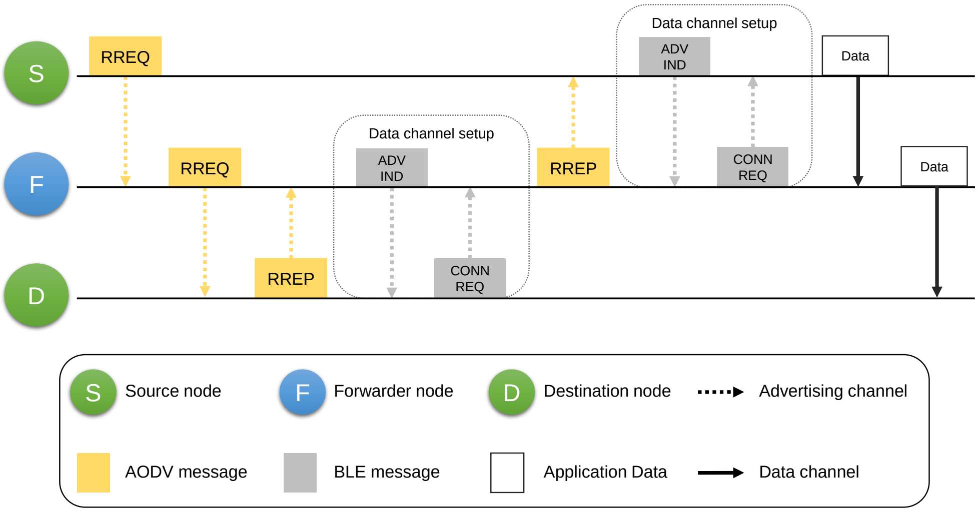


Figure 2.6: End-to-end establishment of an AODV route and BLE connections.

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 Figure 2.6, 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.

2.4.3 Link Quality Metric for BLE Data Channels

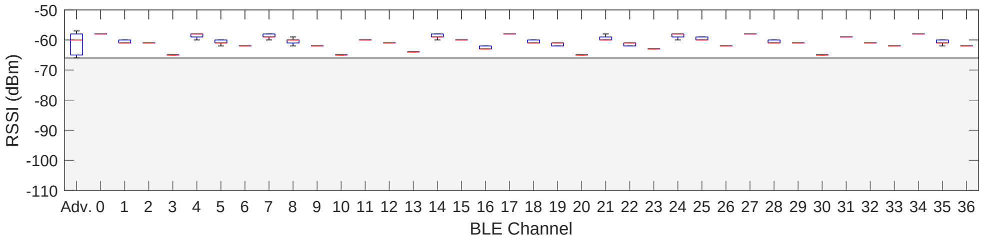
To establish a stable multihop route, it is important to manage a neighbor table that indicates whether each neighbor node's wireless link quality is good enough. Specifically, in AODV routing, the link quality information is passed through the RREQ flooding procedures. When a node receives an RREQ message 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.

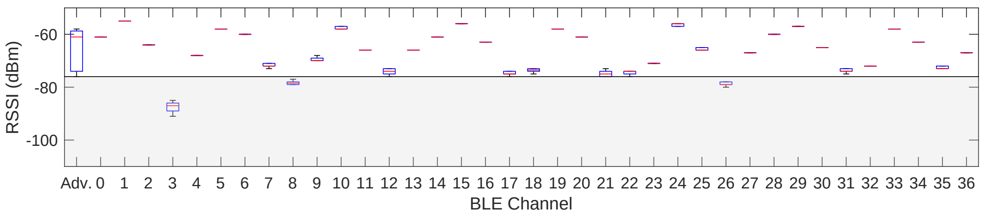
Second, BLE link layer (called controller part) does not have an interface that provides its detailed information for the upper layer (called host part). For example, when a BLE device sends a packet, the host part does not know how many times the packet was retransmitted, which makes obtaining the famous ETX (Expected Transmission Count) metric unfeasible. In addition, when a BLE device receives a packet, the host part can know the packet's RSSI (Received Signal Strength Indicator) 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.

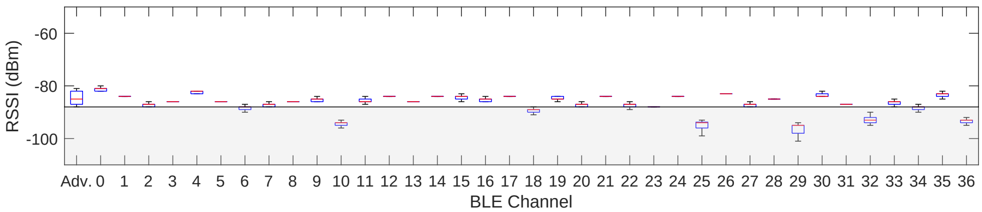
Figure 2.7 represents RSSI values on each BLE channel. Based on the close re-



(a) Good channel



(b) Medium channel



(c) Bad channel

Figure 2.7: 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.

lationship between RSSI and PRR [37], We consider three scenarios: a good channel scenario where all 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.

2.4.4 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 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

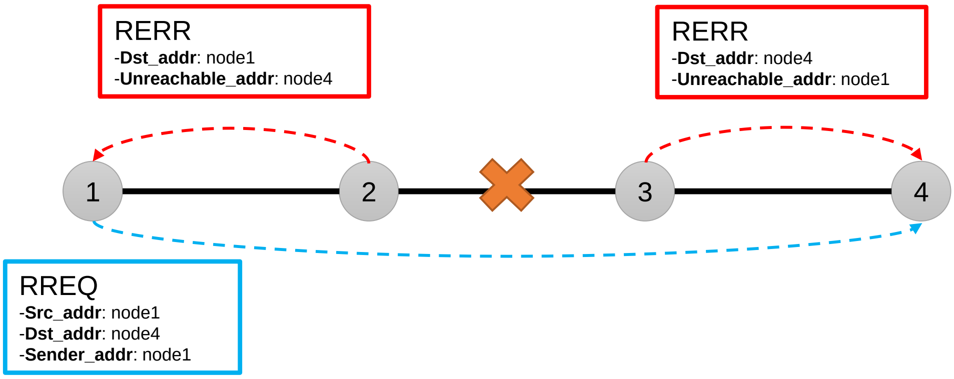


Figure 2.8: Route recovery of MABLE.

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.

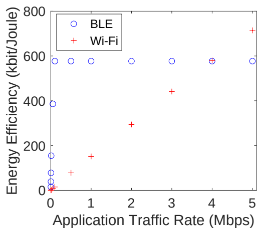
RERR message propagation: Although a node is deleted from the route table, its BLE connection is still left. In contrast to AODV that sends an RERR to the source node when a link breakage is detected, when MABLE detects a broken BLE connection, it sends an RERR to both source and destination nodes (i.e., 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 Figure 2.8. 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.

2.5 Wi-BLE: Wi-Fi Ad-hoc over BLE

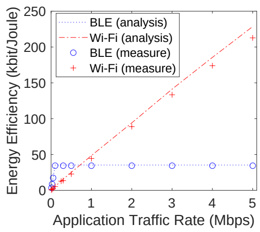
Since a source-destination pair has a BLE-based bi-directional route (always-on BLE), this section describes how to operate Wi-Fi synergistically with BLE: (1) when *Wi-BLE* selects Wi-Fi for data transmission instead of BLE, (2) how *Wi-BLE* constructs a multihop route for Wi-Fi, and (3) how *Wi-BLE*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

2.5.1 Radio Selection

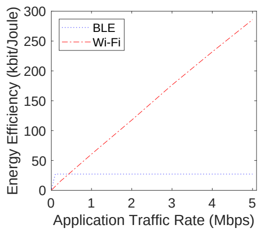
Once a BLE-based route is built for a source-destination pair, the source node selects what radio to use for end-to-end data transmission: Wi-Fi or BLE. We aim to maximize energy efficiency while delivering a given application traffic load. Given that Wi-Fi provides much higher data rate than BLE, using 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.



(a) One hop topology (BLE hop count: 1, Wi-Fi hop count: 1)

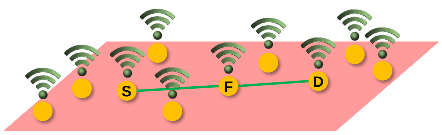


(b) Multi-hop topology (BLE hop count: 6, Wi-Fi hop count: 2.5)

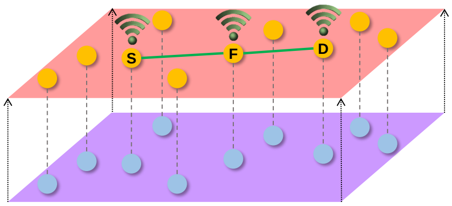


(c) Multi-hop topology (BLE hop count: 7, Wi-Fi hop count: 2)

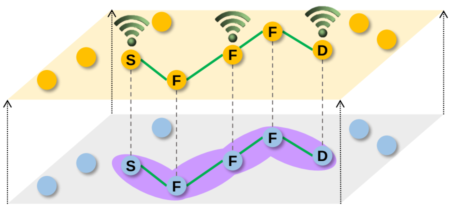
Figure 2.9: Energy Efficiency of Wi-Fi and BLE.



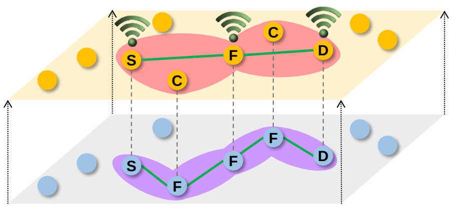
(a) Always-on Wi-Fi



(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

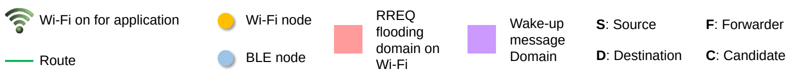


Figure 2.10: Wi-BLE's Wi-Fi routing modes using BLE.

Figure 2.9 shows energy efficiency of BLE and Wi-Fi according to the application traffic rate in three different scenarios. Figure 2.9(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.

Figure 2.9(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 Figure 2.9(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 Figure 2.9(b), we can obtain analytic values similar to the experimental values through one-hop energy efficiency and topology information.

Figure 2.9(c) considers another scenario that reduces the hop distance more significantly (7 to 2) by using Wi-Fi than the scenario in Figure 2.9(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.

2.5.2 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 Figure 2.10(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 Figures 2.10(a), 2.10(b), and 2.10(c).

BLE wake-up radio for Wi-Fi (Mode 1): Figure 2.10(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): Figure 2.10(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): Figure 2.10(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.

2.6 Evaluation

We implement *Wi-BLE* in Linux and evaluate it on real devices. To this end, we configured an indoor testbed as depicted in Figure 2.11, where a total of 31 nodes were deployed with a source (node 14) and a destination (node 6). The route snapshot in Figure 2.11 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.

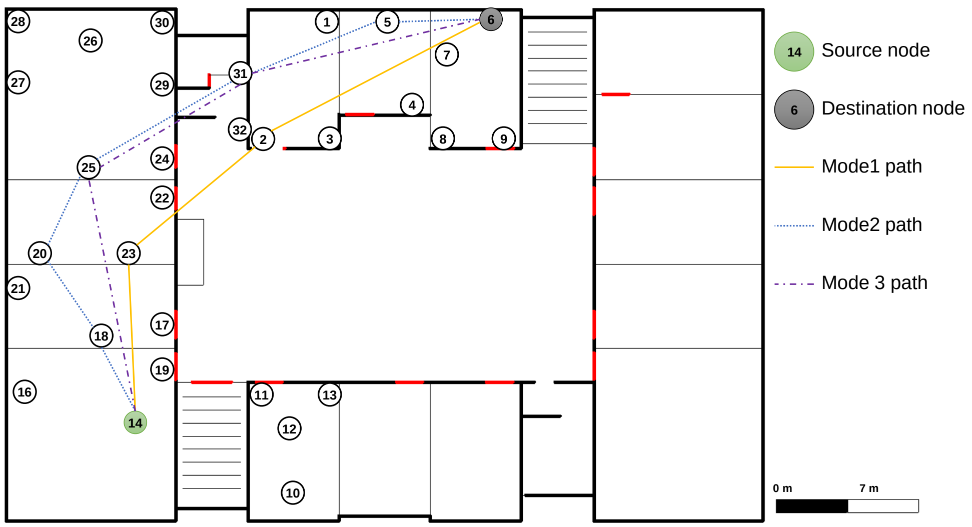


Figure 2.11: Testbed topology and a route snapshot between nodes 14 and 6.

2.6.1 BLE Routing

We first evaluate the performance of BLE network with MABLE. Figure 2.12 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

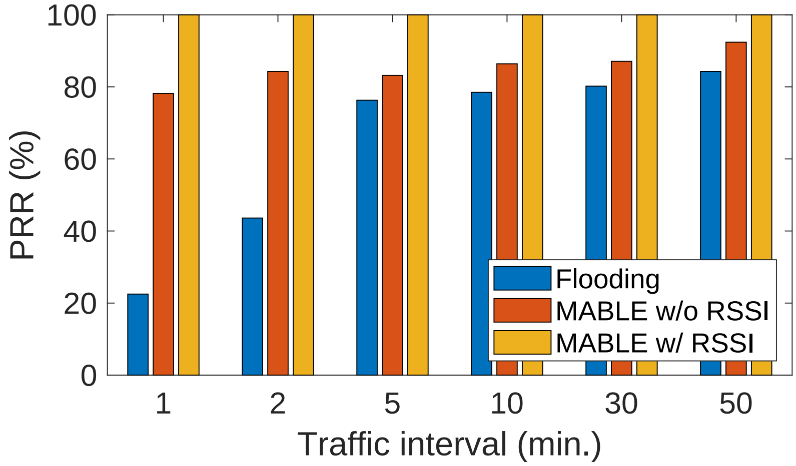
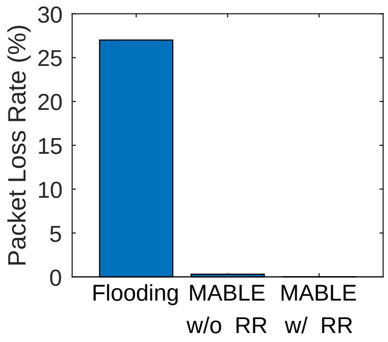


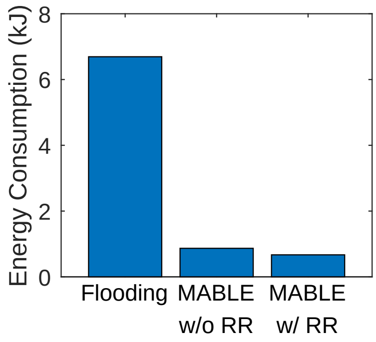
Figure 2.12: Performance of BLE routing schemes.

route is constructed. Figure 2.12 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, 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, Figures 2.13(a) and 2.13(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.



(a) Reliability



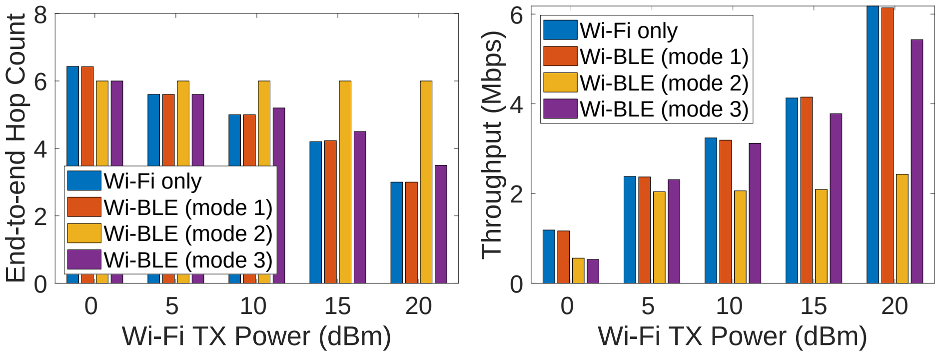
(b) Energy consumption

Figure 2.13: BLE performance with a link breakage when traffic interval is 5 mins.

2.6.2 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.

Figures 2.14(a) and 2.14(b) show end-to-end hop count and throughput according



(a) End-to-end hop count

(b) End-to-end throughput

Figure 2.14: Traffic delivery performance of Wi-Fi network with varying Wi-Fi transmission power.

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

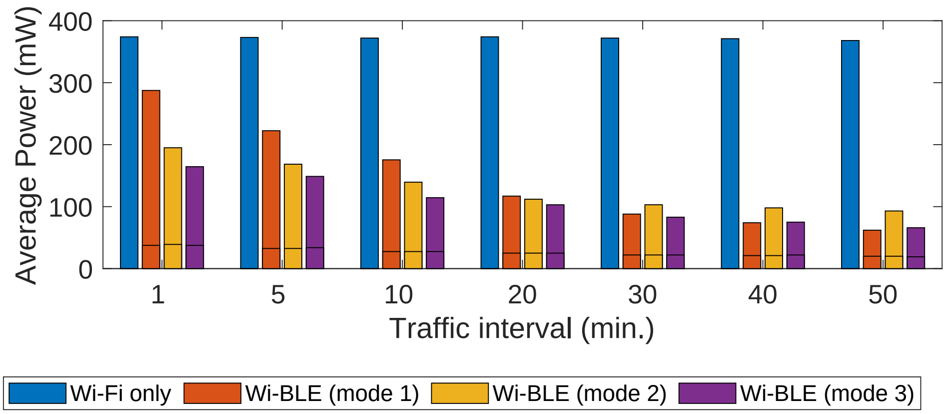


Figure 2.15: Average power consumption of Wi-Fi network according to the traffic interval.

according to traffic interval in Figure 2.15. We used lines in the middle *Wi-BLE* 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 Figure 2.14, 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 Figures 2.14(a) and 2.14(b), the results in Figure 2.15 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.

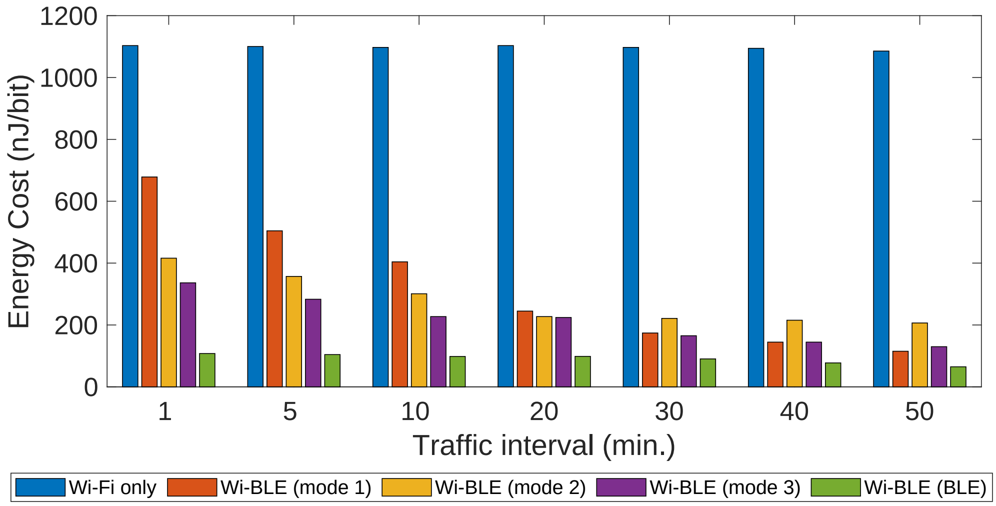
On the other hand, all the three routing modes in *Wi-BLE* only turn on Wi-Fi when necessary, resulting in lower power consumption as traffic interval increases due to less routing overhead. The energy consumed in transmitting control messages in the BLE network is negligible compared to that in transmitting 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.

2.6.3 Radio Selection

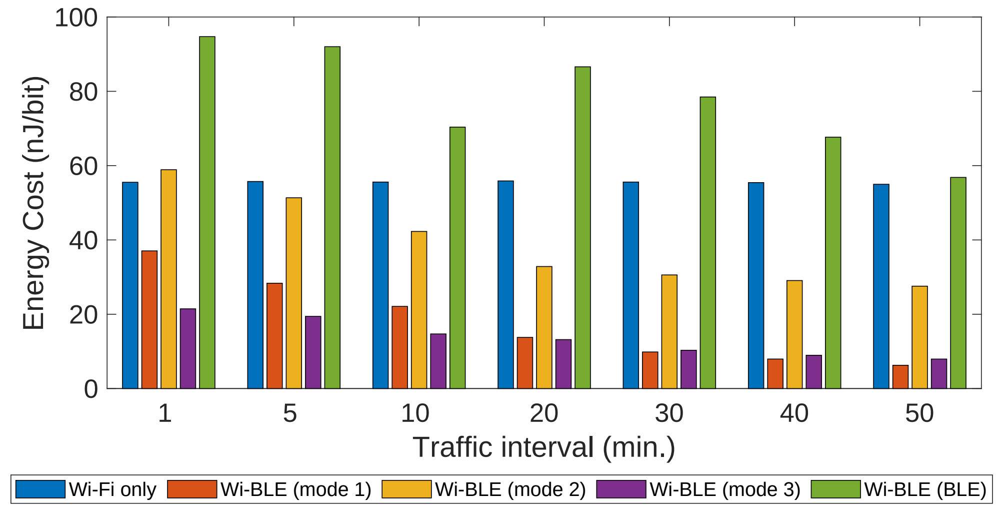
In this section, we evaluate the effectiveness of *Wi-BLE*'s radio selection in terms of energy efficiency (joule per bit), which is a comprehensive metric that includes both throughput and energy consumption. To this end, Figures 2.16(a) and 2.16(b) depict energy efficiency of four *Wi-BLE* options 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.

Figure 2.16(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.

Figure 2.16(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 Figure 2.16(a), *Wi-BLE* (BLE) always provides the worst



(a) Light traffic case (10 kbps)



(b) Heavy traffic case (6 Mbps)

Figure 2.16: Energy efficiency with varying traffic interval when *Wi-BLE*'s data radio is manually selected.

performance in Figure 2.16(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.

2.7 Summary

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 chapter, 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.

Chapter 3

WBC-AFH: Direct Wi-Fi to BLE Communication based AFH

3.1 Introduction

The ISM band is the most commonly used frequency band in wireless communication, such as Wi-Fi, ZigBee, and Bluetooth. As communications using this band become popular, congestion problems due to interference from other wireless communications and collisions between the same wireless communications become more serious. In order to alleviate this collision problem, wireless communication has used conventional approaches such as direct sequence speared spectrum (DSSS) or frequency hopping spread spectrum (FHSS). Among them, Bluetooth based on frequency hopping introduced adaptive frequency hopping (AFH), an advanced channel-hopping method.

AFH detects channels expected to frequently suffer collisions (mainly by Wi-Fi signals) among available channels and excludes these 'bad' channels from the hopping set. With this blacklist-based operation, Bluetooth reduces collisions with other devices by using only clean channels in the hopping set. Moreover, it prevents energy waste or delay increase by reducing retransmission due to collisions. The Bluetooth SIG establishing the Bluetooth standard allows each Bluetooth manufacturer to implement their AFH method without forcing a specific AFH algorithm. Since AFH algorithms of different manufacturers have different specific operations, their performance also shows results.

However, as the ISM bands became more congested with various wireless communication devices, AFH has faced a significant problem. Which channel to use when all channels are congested? For example, if Wi-Fi traffics are transmitted in all 2.4 GHz ISM bands, AFH detects collisions by Wi-Fi on all Bluetooth channels. So it attempts to exclude all Bluetooth channels from the whitelist. Fortunately, most AFH algorithms pre-determine the minimum size of the hopping set, so not all channels are blacklisted. However, blacklisting still has the following two potential problems.

(1) Channel rejoin: Many AFH algorithms mainly focus on how to exclude bad channel but does not care much about how to reuse the excluded channel. Bluetooth devices can easily evaluate currently used channels included in the whitelist, but it is difficult to evaluate excluded channels with blacklisting. Therefore, it is difficult to determine whether to put it back into the hopping set because information on the excluded channels is insufficient. The re-join method can be classified into statistics accumulation with active probing and whitelist reset. Active probing periodically transmits probing packets even in channels that are not currently used. Furthermore, it evaluates the packet transmission results and decides whether to include them in the whitelist again. This method consumes extra energy from the Bluetooth device by generating probing packets even when data transmission is infrequent. The reset-based approach resets the blacklisted channels when the number of Bluetooth channels in the whitelist is lower than a certain number. After that, the Bluetooth device attempts AFH on all channels again. By this method, frequent channel resets occur when Wi-Fi continuously generates traffic on all Wi-Fi channel bands. So it degrades transmission reliability. In conclusion, channel re-join methods lead to additional energy consumption or low reliability of packet transmission.

(2) Channel capacity: When Bluetooth devices using the same AFH algorithm ex-

ist in the same collision domain, it is highly likely to have a similar whitelisted channel set. If the number of neighboring Bluetooth devices is small and the traffic rate is low, sharing the same whitelist is not a problem. However, the higher the number of neighboring Bluetooth devices and the higher the traffic rate, the higher the probability that Bluetooth devices access the same channel in the shared whitelist simultaneously, causing collisions between Bluetooth devices. In particular, in a situation where Wi-Fi traffics congest all Wi-Fi channel bands, the intra-collision problem between neighboring Bluetooth is more severe because the size of the whitelist becomes very small. Paradoxically, the channel determined to be good becomes a bad channel due to Bluetooth collision.

In summary, blacklisting-based AFH may require additional energy consumption to improve packet reliability. Even with such additional energy consumption, preventing collision between neighboring Bluetooth is difficult in a situation where Wi-Fi causes congestion in all ISM bands. Therefore, the transmission efficiency of Bluetooth decreases in such congestion scenarios. With the success of the wireless internet and IoT, as more use of wireless communication devices in daily life, such congestion will become more severe. Therefore, to efficiently use Bluetooth, it is necessary to supplement AFH so that it can operate in such a congested environment.

However, there are several challenges in designing an AFH that is robust against congestion. We summarized the challenges to solving this reduction in transmission efficiency as follows. First, how to accurately evaluate Bluetooth channels with low energy cost? Several well-known metrics, such as received signal strength indication (RSSI), and packet reception ratio (PRR), can evaluate the Bluetooth channel. Unlike other wireless communications that use a single channel, Bluetooth uses multiple channels, so each channel is evaluated with the metrics. Since more than 30 channels are used only for the data channel, the overhead of channel quality evaluation is significant in Bluetooth. Therefore, metrics based on probing cause a substantial energy consumption overhead in Bluetooth. Meanwhile, a metric based on passive statistics may be inaccurate depending on changes in the channel environment. Therefore, a method for accurately evaluating each Bluetooth channel with low overhead is required. Second, how to minimize collision in a limited Bluetooth channel? For successful channel hopping, Bluetooth requires some minimum channel set size. However, when Wi-Fi traffic is congested, the number of channels free from interference can be minimal. In particular, since a channel free from Wi-Fi interference is shared by neighboring Bluetooth, the collision between Bluetooth devices may intensify as the number of such channels decreases. Therefore, a method that efficiently utilizes a limited number of Bluetooth channels in the ISM band is required to minimize Wi-Fi interference and neighboring Bluetooth interference.

For these challenges, we will improve the transmission efficiency of Bluetooth, especially Bluetooth Low Energy (BLE), with the following approach.

Evaluating channel quality based on Cross Technology Communication (CTC): We use the latest CTC technology for accurately evaluating channel quality with low energy costs. Utilizing the latest CTC, especially PHY level CTC, Wi-Fi transmits information to the BLE device so that the BLE device can evaluate the collision probability and congestion level of the channel with small overhead.

Optimizing the size of the channel hopping set: We propose a method to optimize the number of channels that minimize collision by using Wi-Fi information and neighboring BLE node information. BLE can improve transmission efficiency with a channel set size optimized to reduce interference.

Weighting channel selection ratio with channel quality: We modify the existing AFH that uniformly uses channels in the channel set. By weighting each channel according to channel quality, BLE can use the limited number of channels more efficiently.

Based on the approaches, we make the following contributions in this chapter. First, we proposed a new adaptive frequency hopping algorithm named *WBC-AFH*. *WBC-AFH* exploits CTC to evaluate each BLE channel with low energy cost and calculates the optimal channel set size and weight for each channel. Second, we evaluate performance of *WBC-AFH* with simulations and prototype implementation. We implemented *WBC-AFH* in a real BLE device to evaluate it on a small scale and designed a simulator for evaluation on a large scale. Lastly, we show significant improvement in the reliability of BLE with *WBC-AFH*. *WBC-AFH* improved the reliability of BLE in any environment compared to conventional AFH of various methods.

The structure of this chapter is as follows. Section 3.2 introduces background of Adaptive Frequency Hopping. Section 3.3 describes the proposed schemes, *WBC-AFH*. Section 3.4 provides evaluations of *WBC-AFH* and a comparison to conventional AFH algorithms. We discuss the future work of *WBC-AFH* in 3.5. Finally, we summarize our work in 3.6

3.2 Background

3.2.1 Frequency hopping in BLE

Devices using the 2.4GHz ISM band require the ability to withstand interference as most countries approve this band for unlicensed use. One of the mechanisms that Bluetooth devices use to achieve this resilience is frequency hopping. BLE's connection-based data channel and connectionless-based secondary advertising channel use one channel for each event. The channel selection algorithm determines the channel for the next connection event for each connection event. There are two types of channel selection algorithms. This channel selection is pseudo-random. Randomness minimizes overlap of channel occupancy by reducing the likelihood of channel collisions. There are two types of channel selection algorithms, but both use channels uniformly.

Adaptive frequency hopping

Adaptive frequency hopping extends the frequency hopping of BLE by dynamically removing channels from the hopping sequence to adapt to changes in the channel environment. A Channel map indicates channels that can be used by each event and channels that cannot be used as a blacklist. The Bluetooth specification [38] defines the need for AFH and various paths to obtain information for AFH, but does not define how such information would be used for AFH. The specific AFH algorithm depends on the implementation of the BLE manufacturer. So different implementations have different AFH mechanisms by BLE manufacturers.

Link quality evaluation

BLE vendor can use a different metric to evaluate the quality of a different BLE channel. The most commonly used one is to evaluate the RSSI [39] in the idle state in the corresponding channel and measure the degree of interference that can potentially occur in the corresponding channel. Another metric is packet reception ratio (PRR) [40], indicating the ratio of successfully transmitted packets receiving an ACK. Although PRR takes time to obtain an accumulated value, it enables a more accurate channel quality evaluation than RSSI.

Channel classification

According to link quality evaluation, AFH classifies whether to enable or disable each BLE channel. BLE has a channel map of 37 bits, and each bit corresponds to one BLE data channel. An enabled channel is marked as 1, and a disabled channel is marked as 0. The disabled channel in the hopping sequence is replaced with one of the enabled channels in the channel map. With specific link information of each channel, the algorithm for determining which channel to enable also differs depending on the BLE vendor. We summarize these algorithms in two main types.

Threshold based classification Algorithms of most BLE vendors define a fixed threshold for a specific link quality metric. AFH blocks all channels with link quality lower than the threshold in this threshold-based classification. Some advanced threshold-based algorithms blacklist related channels all at once with a bit of probing.

This method has the advantage of being able to guarantee the quality of all enabled channels in the channel set. However, a too strict threshold may cause starvation problems that reduce the number of enabled channels. Therefore, an appropriate threshold value is required, but setting a threshold value suitable for environmental changes is challenging.

Ranking based classification A more robust classification method for the channel starvation problem is based on ranking according to link quality. This classification method enables a predetermined number of channels with the best quality among all BLE channels. A reasonable channel set size overcomes the starvation problem and adapts well to a dynamic channel environment. However, it does not guarantee the quality of each channel. If the channel set size is too large, even channels with bad quality may be included in the channel set, thereby reducing transmission efficiency. On the other hand, if the channel set size is too small, even channels of sufficiently good quality can be excluded. Therefore, it is possible to reduce robustness to channel change or increase the probability of collision between neighboring BLE nodes. As a result, a channel set size parameter should be set appropriately for ranking-based classification.

In summary, we point out that both classifications require appropriate parameter settings. Our *WBC-AFH* optimizes the necessary parameters such as channel set size based on the information provided around the BLE device without any previous parameter setting. Also, we point out that both classifications determine the enable of channels as binary decisions and uniformly select the channels in the whitelist. Our *WBC-AFH* follows a weighted probability distribution, not binary channel selection.

3.2.2 Cross Technology Communication

This section describes CTC, a technique exploited in the proposed scheme. Cross-technology communication (CTC) is a technology that deals with direct packet exchange between other wireless communications. In order to overcome different modulations and operations, the previous CTC has taken a strategy of transmitting symbols by artificially manipulating the packet itself. This strategy is called packet-level CTC, and methods using packet length [41], [42], timing [43], [44], and energy level [45], [46]. In this packet-level CTC, transmission efficiency is very low because one packet is mapped to one symbol. Also, since various artificial manipulations had to be applied to the packet, there is a limitation that it is difficult to use in an actual COTS device.

However, with the PHY-level CTC emerging as a new CTC paradigm, the CTC technique has changed rapidly. PHY-level CTC exploits imitation of different modulation called symbol emulation. Symbol emulation provides a mapping of a symbol encoded with a specific modulation (especially sensitive to phase shift) to a symbol decoded with another modulation. Compared to packet-level CTC, PHY-level CTC shows significantly higher throughput. Also, since most PHY-level CTCs use simple symbol mapping-based emulation, there is not necessary to modify the hardware or firmware. With higher throughput and lower hardware dependencies, PHY-level CTC is expanding its applications.

WEBee [47] is the first PHY-level CTC. It enables direct communication from Wi-Fi to ZigBee with emulation of QAM to QPSK. WeBee shows that commodity ZigBee receiver can decode RX signal from commodity Wi-Fi NIC card to a legitimate ZigBee packet. Also it achieve 126 Kbps with high reliability in noisy environment, a throughput about 16,000 times that of packet-level Wi-Fi to ZigBee CTC [41].

Another PHY-level CTC is BlueBee [48]. BlueBee enables direct communication from BLE to ZigBee with emulation of GFSK to QPSK. In BlueBee, BLE transmitter is transparent to the ZigBee reciver, and vice versa. BlueBee emulates legitimate ZigBee packet by encapsulating BLE symbols mapped with ZigBee symbols within BLE packet. Due to the similarity between GFSK and QPSK modulation, BlueBee can achieve the maximum data rate of ZigBee, 250 kbps at more than 99% accuracy. Its throughput is 10,000x higher than the BLE to ZigBee packet-level CTC [43]. BlueBee also showed its feasibility by showing that a smartphone equipped with BLE controls a smart light bulb equipped with ZigBee.

Another PHY-level CTC, XBee [49] provides direct communication from ZigBee to BLE. PHY-level CTC increases the number of radio pairs for direct communication while improving the performance of the existing pair [50]–[54].

ECC [55] shows an useful application using CTC. With WeBee, ECC mitigates cross-technology interference (CTI) by allowing Wi-Fi to inform ZigBee directly. In ECC, one Wi-Fi device provides an idle slot from other Wi-Fi packets through a Clear to Send (CTS) packet. Then it transmits cross-technology CTS to ZigBee, informing that there is a whitespace where ZigBee can communicate interference-free. ECC shows that CTC can be utilized successfully for coordination between different communications.

Like ECC, the *WBC-AFH* utilizes PHY-level CTC, specifically Wible [56], which enables Wi-Fi to BLE CTC to reduce interference between communication technologies. Through ECC, the proposed can exploit direct communication of high data rates and avoid unnecessary energy consumption in BLE.

3.3 Proposed AFH

The section introduces the system of *WBC-AFH*. We defines two modules to be added to Wi-Fi AP in *WBC-AFH*. The first module is for channel statistics. It records the busy time of channel used by the Wi-Fi network. Through this, it statistically calculates the idle time ratio of the corresponding Wi-Fi channel. The other module creates CTC packets for BLE. It periodically transmits the information of idle time ratio in the form of CTC packets that BLE can hear.

Also, we modify the BLE master device to periodically generate a packet informing the neighboring BLE master of its existence and the connection interval (CI) value.

Based on these modifications, the BLE master receives the idle time ratio sent by Wi-Fi and the CI value of the neighboring BLE node. With this information, the *WBC-*

AFH optimizer calculates the size of the hopping set and the weight of each channel.

3.3.1 CTC based informing

This section describes how Wi-Fi radio generates CTC packets for BLE. We use Wible to generate packets that Wi-Fi can decodable by BLE. One of the critical observations of PHY level CTC is that low-cost IoT devices do not use modulation based on time domain waveform but are based on modulation sensitive to phase change. BLE uses GFSK modulation, but this GFSK modulation can be re-mapped with a sequence of positive and negative phase changes. Due to these characteristics, the Wi-Fi radio uses symbol transition to generate BLE decodable packets.

Wi-Fi uses an 802.11b signal to generate CTC packets. 802.11b uses DSSS, bit 1 is spread to 10110111000, and bit 0 is spread to 01001000111. This chip sequence is converted into a phase shift sequence through DQPSK modulation and into an analog signal through a DAC converter.

When the BLE radio succeeds in receiving the emulated preamble using the Wi-Fi signal, the BLE radio recognizes the emulated phase shift sequence as a valid BLE packet and attempts demodulation. GFSK modulation of BLE attempts demodulation with BLE bit 1 when there is a positive phase change and with BLE bit 0 when there is a negative phase change. According to Wible's observation, when the Wi-Fi bit sequence is 10 or 01, BLE generates a negative phase shift, and when the Wi-Fi bit is 11 or 00, BLE generates a positive phase shift. This observation surprisingly means that the phase shift sequence of DQPSK modulation of Wi-Fi can be mapped to the same sequence as the BLE phase shift in the form shown in the figure. For example, if Wi-Fi transmits 11, 01, and 11 symbol sequences, BLE demodulates 1, 0-bit sequences according to the phase shift between symbols.

Through this Wi-Fi symbol sequence to BLE bit sequence mapping, Wi-Fi can obtain a symbol sequence stream that BLE can receive and generate a corresponding packet. The emulated packet is transmitted in the Wi-Fi 4 channel overlapping the BLE

38 channel so BLE could periodically scan the advertising channel.

3.3.2 Weighted channel select

With the idle time ratio $r_{idle,i}$ of each Wi-Fi channel informed by Wi-Fi to BLE CTC, *WBC-AFH* calculates the probability of collision-free with Wi-Fi traffic expected in each BLE channel as equation 3.1.

$$p_{free,c} = \prod^{i \subset S} (r_{idle,i}) \tag{3.1}$$

At this time, *S* means a set of overlapped Wi-Fi channels with BLE channel index c.

WBC-AFH mainly attempts to select a BLE channel with a high probability of collision-free. It opportunistically attempts to use a channel with a relatively high probability of collision with a low weight. The weight of channel *c* is obtained by function of $p_{free,c}$ as

$$w_n = p_{free,c}^{\alpha} \tag{3.2}$$

where the value α is a parameter that adjusts the weight gap between the Wi-Fi free channel and the congested channel. When α is set to a large value, AFH uses the safest channel as much as possible and reduces opportunistic use of other channels. When small α value, AFH uses the entire channel close to uniform.

Normalized weight obtained by function as

$$p_{weight,c}(\alpha) = \frac{w_c}{\sum_{i=0}^{36} w_i} = \frac{p_{free,c}^{\alpha}}{\sum_{i=0}^{36} p_{free,c}^{\alpha}}. \tag{3.3}$$

The weights are given so that the access weight is proportional to the collision free probability. *WBC-AFH* set the probability of selecting a channel from the hopping set to $p_{weight,c}(\alpha)$.

Figure 3.1 shows an example of channel selection probability distribution by weighted channel selection.

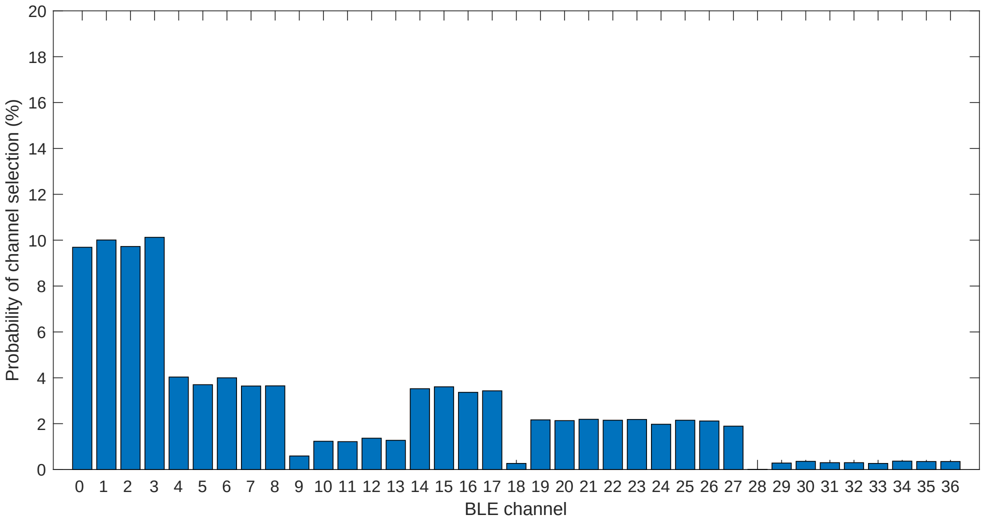


Figure 3.1: An example of channel selection probability distribution of weighted channel selection.

3.3.3 Hopping set size optimization

In the case of the ranking-based channel classification method, AFH uses the predefined fixed size of the hopping set. AFH has better adjust the hopping set's size n_{ch} appropriately according to the surrounding interference situation. Suppose the size of the hopping set is too large. In that case, AFH includes unnecessarily interfered channels in the hopping sequence. Thereby overall packet transmission reliability decreases. Conversely, if the size of the hopping set is too small, the probability of collision increases by selecting similar channels as the neighboring BLE. Therefore, selecting the size n_{ch} of the hopping set is crucial considering both Wi-Fi and interference and neighboring BLE interference.

WBC-AFH calculates the optimal size of the hopping set to minimize collision based on direct CTC information from Wi-Fi and information from neighboring BLE. For now, we introduce naive hopping set size optimization. Int naive optimization, each channel is uniformly selected from the channel map by pseudo-random.

The probability P_{BLE_free} that a BLE node transmits a packet without collision by neighboring BLE is given as

$$P_{BLE\_free}(n_{ch}) = [1 - \frac{2}{n_{ch}} \times \frac{t_{BLE\_TX}}{CI}]^{N_{BLE\_node}} \tag{3.4}$$

where CI is the connection interval and N_{BLE_node} is the number of the neighboring BLE obtained by advertising, and t_{BLE_TX} is the air time duration of a BLE packet given by packet length and data rate of BLE. P_{BLE_free} increases as the channel set size value of n_{ch} increases.

The probability P_{Wifi_free} that a BLE node transmits a packet without collision by Wi-Fi networks is given as

$$\begin{aligned} P_{Wifi\_free}(n_{ch}) = (\frac{1}{n_{ch}}) \times \sum_{i=1}^{n_{ch}} (1 - \hat{p}_{free,i}). \\ (where\ \ \hat{p}_{free,1} > \hat{p}_{free,2} > ... > \hat{p}_{free,n_{ch}-1} > \hat{p}_{free,n_{ch}}) \end{aligned} \tag{3.5}$$

Note that $\hat{p}_{free}$ is sorted sequence of p_{free}. P_{Wifi_free} decreases by gradually adding channels with low link quality as the channel set size value of n_{ch} increases.

Finally, the probability P_{free} that a BLE node successfully transmit a packet without both Wi-Fi collision and the neighboring BLE collision is given as

$$P_{free}(n_{ch}) = P_{BLE\_free}(n_{ch}) * P_{Wifi\_free}(n_{ch}). \tag{3.6}$$

We formulate an optimization problem to find an optimal n_{ch} as

$$\begin{aligned} \underset{\mathrm{n_{ch}}}{\text{maximize}} \quad & P_{free}(n_{ch}) \\ \text{subject to} \quad & 0 < n_{ch} \leq 37. \end{aligned} \tag{3.7}$$

This optimization problem is not convex problem. However, n_{ch} is integer variable, so we can find an optimal n_{ch} value in fixed time.

With optimized value, AFH enables n_{ch} channels with the best link quality P_{free} in naive hopping set size optimization.

Figure 3.2 shows an example of channel selection probability distribution by naive hopping set size optimization.

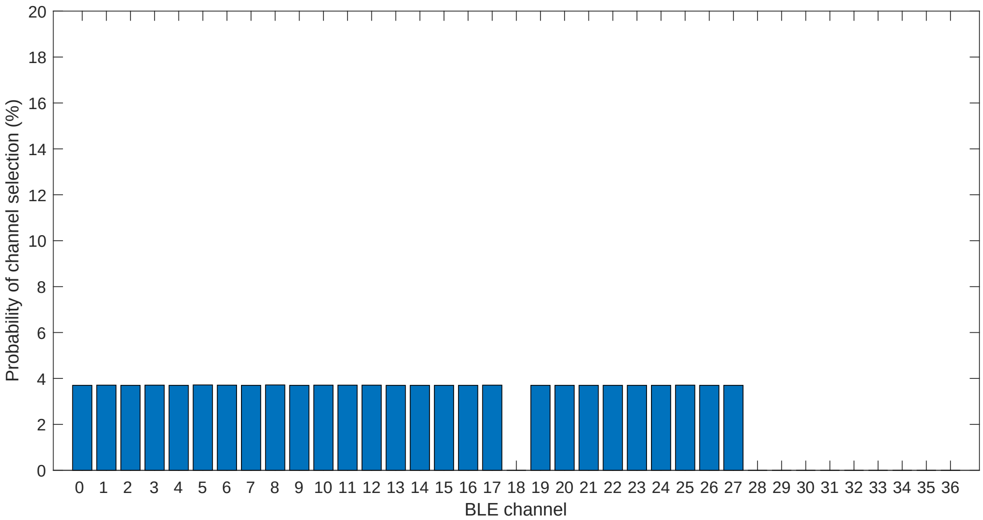


Figure 3.2: An example of channel selection probability distribution by naive hopping set size optimization.

3.3.4 *WBC-AFH*

Here, we describe the final proposed scheme combining the previously described weighted channel selection and hopping set size optimization.

The naive optimization described above in 3.3.3 assumes that each channel is selected from the channel map with uniform probability distribution. However, if a channel *j* is chosen with different probability q_j in the channel map, the probability functions in 3.3.3 can be generalized as follows.

The probability P_{BLE_free} that a BLE node transmits a packet without collision by neighboring BLE is modified as

$$P_{BLE\_free}(\widetilde{q}) = \sum_{k=1}^{37}[q_k \times (1 - 2q_k \times \frac{t_{BLE\_TX}}{CI})^{N_{BLE\_node}}] \tag{3.8}$$

where $\widetilde{q}$ is a vector from q probability value that indicates selection probability of each BLE channel from channel 0 to 36.

The probability P_{Wifi_free} that a BLE node transmits a packet without collision by Wi-Fi networks is modified as

$$P_{Wifi\_free}(n_{ch}, \widetilde{q}) = \sum_{i=1}^{n_{ch}}[q_i \times (1 - \hat{p}_{free,i})]. \tag{3.9}$$

$$(where\ \ \hat{p}_{free,1} > \hat{p}_{free,2} > ... > \hat{p}_{free,n_{ch}-1} > \hat{p}_{free,n_{ch}})$$

Finally, the probability P_{Wifi_free} that a BLE node transmits a packet without collision by Wi-Fi networks is modified as

$$P_{free}(n_{ch}, \widetilde{q}) = P_{BLE\_free}(\widetilde{q}) * P_{Wifi\_free}(n_{ch}, \widetilde{q}). \tag{3.10}$$

We can formulate an optimization problem to find an optimal n_{ch} and $\widetilde{q}$ as

$$\begin{aligned} \underset{\mathrm{n_{ch}}, \widetilde{\mathrm{q}}}{\text{maximize}} \quad & P_{free}(n_{ch}, \widetilde{q}) \\ \text{subject to} \quad & 0 < n_{ch} \leq 37 \\ & 0 \leq q_1 \leq 1 \\ & 0 \leq q_2 \leq 1 \\ & ... \\ & 0 \leq q_{37} \leq 1 \end{aligned} \tag{3.11}$$

where n_{ch} is integer variable, but element q of vector $\widetilde{q}$ is real variable from 0 to 1. However, since this optimization formula has a very high dimension with 37 real variables, it is difficult to solve the optimization problem in practice.

Here, we simplify the optimization problem by mapping each q_j to the $p_{weight,j}$ proportionally as

$$q_j = p_{weight,j}(\alpha). \tag{3.12}$$

In other words, we set selection weights of the BLE channels according to the degree of Wi-Fi interference.

Accordingly, each collision probability is simplified as follows.

The probability P_{BLE_free} that a BLE node transmits a packet without collision by neighboring BLE is simplified as

$$\begin{gathered} P_{BLE\_free}(n_{ch}, \alpha) = \sum_{k=1}^{37} [p_{weight,k}(\alpha) \times (1 - p_{weight,k}(\alpha) \times \frac{t_{BLE\_TX}}{CI})^{N_{BLE\_node}}]. \\ (where \;\; \hat{p}_{weight,1} > \hat{p}_{weight,2} > ... > \hat{p}_{weight,n_{ch}-1} > \hat{p}_{weight,n_{ch}}) \\ (where \;\; \hat{p}_{weight,n_{ch}+1} = \hat{p}_{weight,n_{ch}+2} = ...\hat{p}_{weight,37} = 0) \end{gathered} \tag{3.13}$$

Note that $\hat{p}_{weight}$ is sorted sequence of p_{weight} and $\hat{p}_{weight}$ of 37-n_{ch} channels with the worst link quality is set to 0. This means that only n_{ch} BLE channels are enabled, and each enabled channel is selected with weighted probability.

The probability P_{Wifi_free} that a BLE node transmits a packet without collision by Wi-Fi networks is simplified as

$$\begin{gathered} P_{Wifi\_free}(n_{ch}, \alpha) = \sum_{i=1}^{n_{ch}} [p_{weight,i}(\alpha) \times (1 - \hat{p}_{free,i})]. \\ (where \;\; \hat{p}_{free,1} > \hat{p}_{free,2} > ... > \hat{p}_{free,n_{ch}-1} > \hat{p}_{free,n_{ch}}) \\ (where \;\; p_{weight,1} > p_{weight,2} > ... > p_{weight,n_{ch}-1} > p_{weight,n_{ch}}) \end{gathered} \tag{3.14}$$

Note that both p_{free} and p_{weight} are fixed values obtained via Wi-Fi CTC.

Finally, the probability P_{Wifi_free} that a BLE node transmits a packet without collision by Wi-Fi networks is simplified as

$$P_{free}(n_{ch}, \alpha) = P_{BLE\_free}(n_{ch}, \alpha) * P_{Wifi\_free}(n_{ch}, \alpha). \tag{3.15}$$

As a result of simplification, given probability P_{Wifi_free} has only two integer variables, n_{ch} and α.

Final optimization problem to find an optimal n_{ch} and α is formulated as

$$\begin{aligned} \underset{n_{ch},\alpha}{\text{maximize}} \quad & P_{free}(n_{ch}, \alpha) \\ \text{subject to} \quad & 0 < n_{ch} \leq 37 \\ & 0 \leq \alpha \leq N \end{aligned} \tag{3.16}$$

where *N* is a parameter that controls the size of the dimension of optimization. As a result, *WBC-AFH* solves optimization problem within $37 \times N$ iteration. With optimized n_{ch} and α values, *WBC-AFH* enables n_{ch} channels with the best link quality P_{free} and selects each channel with the weighted probability P_{weight} obtained by P_{free} and weight gap parameter α.

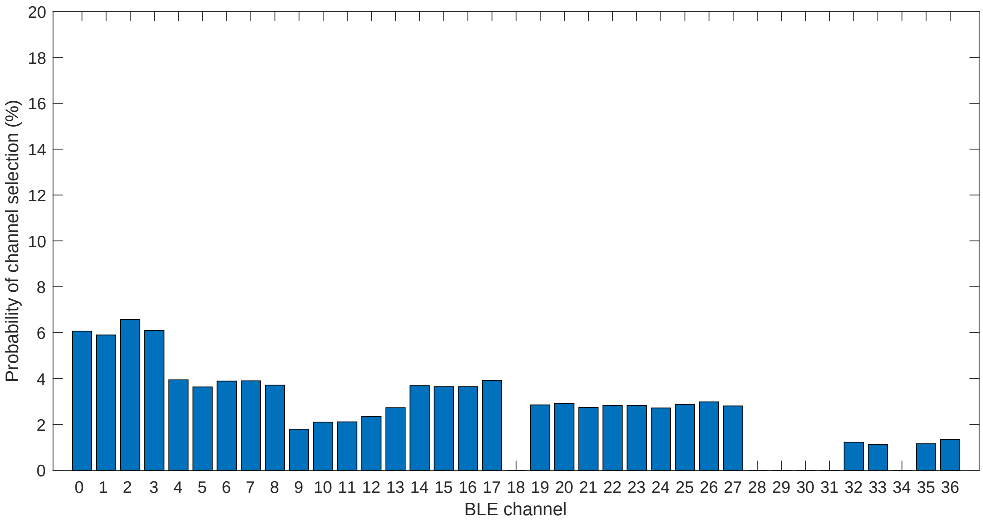


Figure 3.3: An example of channel selection probability distribution of *WBC-AFH*

Figure 3.3 shows an example of channel selection probability distribution of *WBC-AFH*.

3.4 Evaluation

3.4.1 Setup

We evaluated the performance of *WBC-AFH* through actual prototype implementation and simulation. Prototype BLE was implemented through nordic's nrf820 device and Zephyer OS. Wi-Fi uses Atheros AR9271 chipset. Simulation was implemented based on matlab.

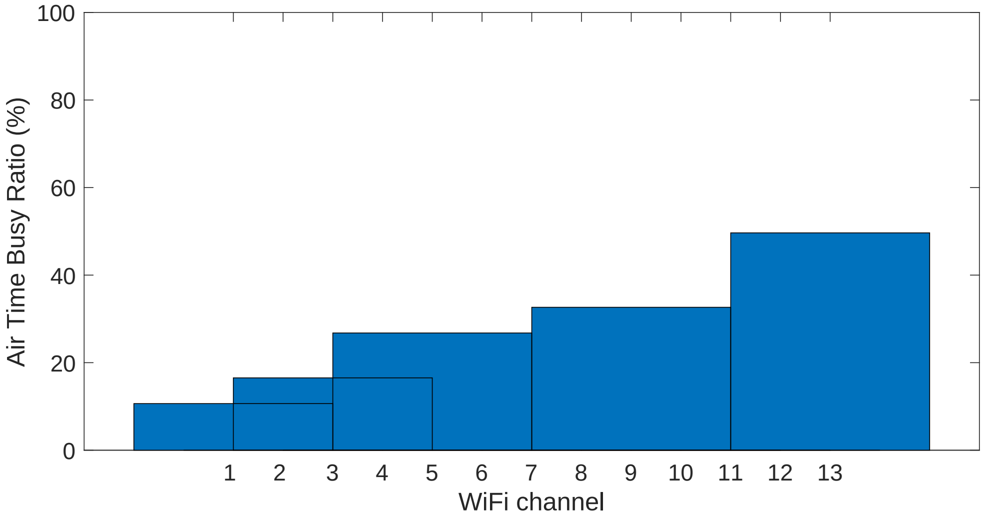


Figure 3.4: Traffic intensity by Wi-Fi channel.

As for Wi-Fi, 5 units operated as APs in the experimental environment, and packets were generated in Wi-Fi channels 1, 3, 5, 9, and 13, respectively. The idle time ratios of each Wi-Fi channel are set to 10%, 16%, 27%, 33%, 50%, respectively. We set the Wi-Fi APs to generate 802.11n traffic. Figure 3.4 shows the traffic settings for each Wi-Fi channel in our experiment, and figure 3.5 shows expected PRR of BLE channels by

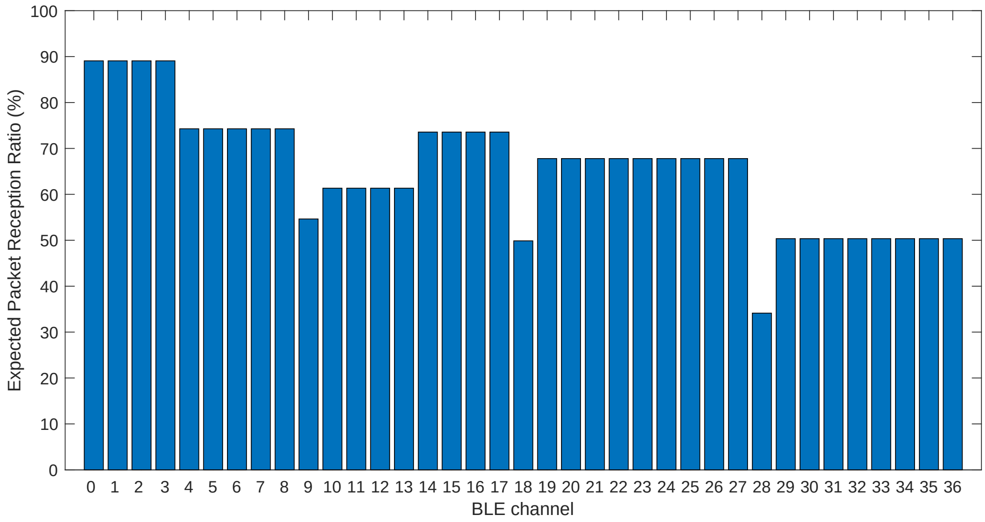


Figure 3.5: Expected PRR in each BLE channel according to Wi-Fi traffic.

Wi-Fi interference.

The BLE node was tested with 5 pairs in the prototype experiment, and up to 100 pairs were changed in the simulation. We set the BLE masters to use 1 Mbps data rate modulation.

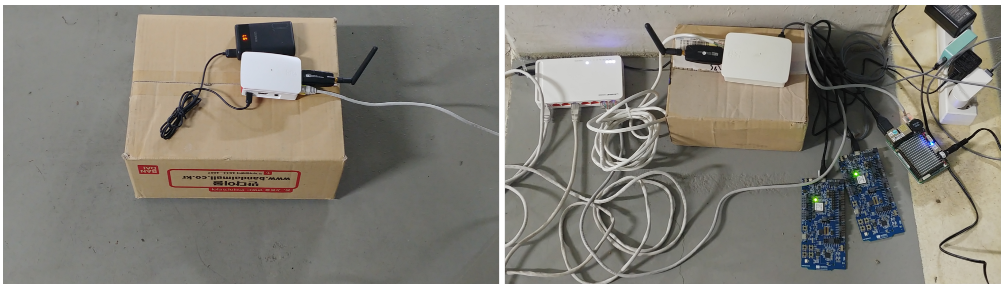


Figure 3.6: Experiments in the parking lot.

We experimented with implementation in an underground parking lot environment with no external Wi-Fi signal we could not control. Figure 3.6 shows an example of an experimental environment.

We selected the following comparison methods to check the performance of our proposed *WBC-AFH*. First, the weight-based AFH and hopping set size optimization AFH methods to correspond to our proposed method's submodules. For weight-based AFH, the α value, a parameter of channel access weight deviation, is set to 4.

Second, we selected the typical conventional AFH techniques, ranking-based AFH and threshold-based AFH, as comparison methods. The hopping set size N of the ranking-based AFH was set to 15, which is the minimum number of channels recommended by the FCC, and the PRR threshold of the AFH based on the threshold was set at 90%.

Finally, we compared the proposed method with a baseline that does not use AFH and consistently uses all BLE channels as channel maps.

3.4.2 Robustness

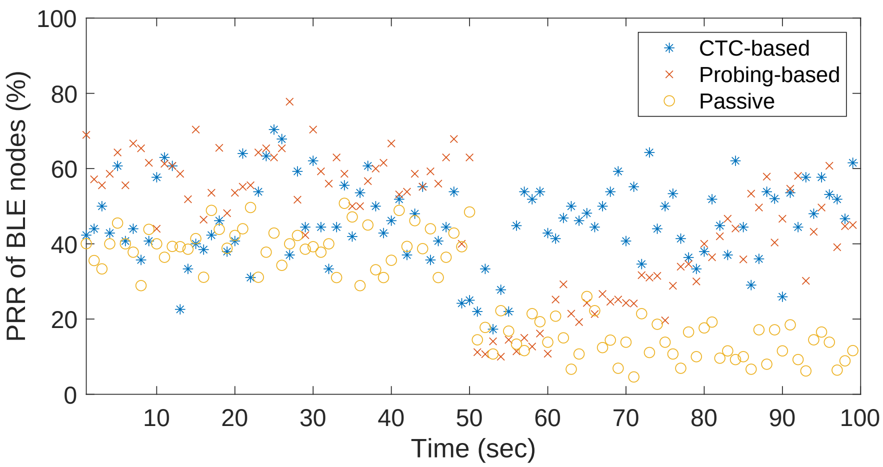


Figure 3.7: Fast channel map update with CTC direct informing.

Figure 3.7 shows the change in PRR of AFH using CTC informing, AFH based on probing, and passive AFH when the Wi-Fi environment is moved to another place due to mobility in BLE. In the case of CTC-based AFH, the channel map was updated by quickly reflecting the channel evaluation based on direct information delivery of Wi-Fi. On the other hand, in the AFH, based on probing, it can be confirmed that there is a delay until the PRR is stabilized. In conclusion, direct information transfer using CTC improves the robustness of BLE in a dynamic situation.

3.4.3 Reliability

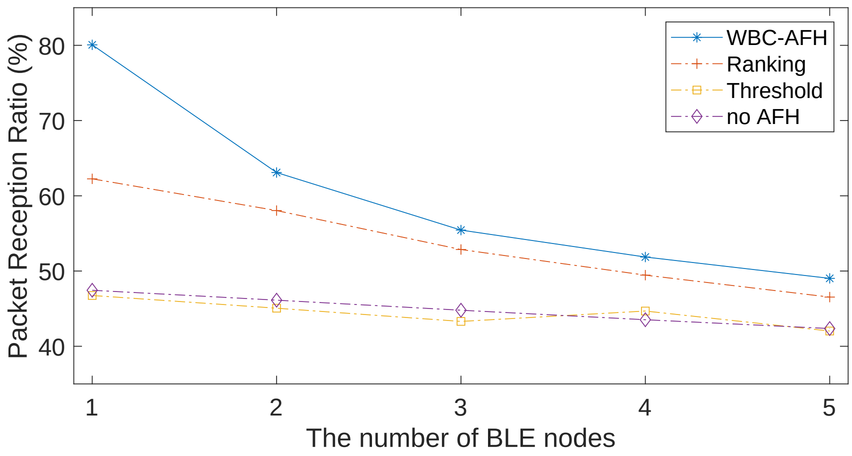


Figure 3.8: Reliability comparison by implementation based experiments.

The figure 3.8 shows the reliability of each frequency hopping method when the number of BLE nodes changes with implementation based experiments. In this experimental environment, we set the BLE node to have a connection interval of 10 msec and an application traffic rate of 600 kbps. In all cases, the packet reception ratio of the proposed scheme shows the best reliability performance. If the BLE traffic is sparse, the room to be optimized is large. Therefore, especially *WBC-AFH* shows good performance when the number of BLE nodes is small. As a result, the proposed method can improve performance in the congested ISM band in small-scale nodes.

The figure 3.9 shows the reliability of submodules on the proposed scheme with previous experiments. Optimizing hopping set size is adequate when the number of

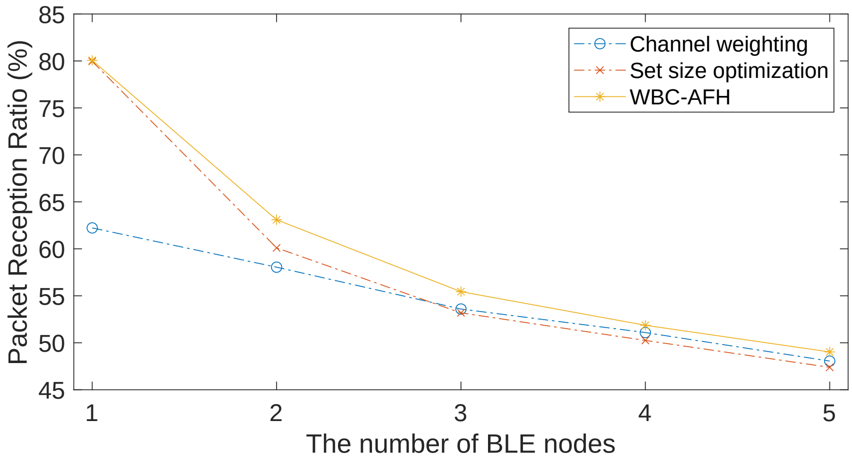


Figure 3.9: Reliability of submodules by implementation based experiments.

BLE nodes is small. However, by increasing the number of BLE nodes, the collision probability of neighbor BLE increases. Therefore, we can say that channel weighting is more effective when that case. As a result, by combining the two submodules, we confirmed that the proposed method always shows good reliability performance.

The figure 3.10, 3.11, 3.12, and 3.13 shows the simulation results of reliability of each frequency hopping technique when the BLE traffic rate changes as the connection interval changes. In all cases, the packet reception ratio of the proposed AFH methods shows the best reliability. So we confirm that the proposed method improves performance in the congested ISM band even in large-scale BLE nodes.

When AFH is not used, it always shows poor performance PRR results. However, one noteworthy point is that the threshold-based AFH also performs almost the same without AFH. If all BLE channels show lower performance than the predefined threshold, the threshold-based AFH no longer works properly. As a result, this point shows the limitations of threshold-based AFH.

The submodule of channel weight and ranking-based AFHs perform better than threshold-based AFHs in an environment with low BLE congestion probability and a

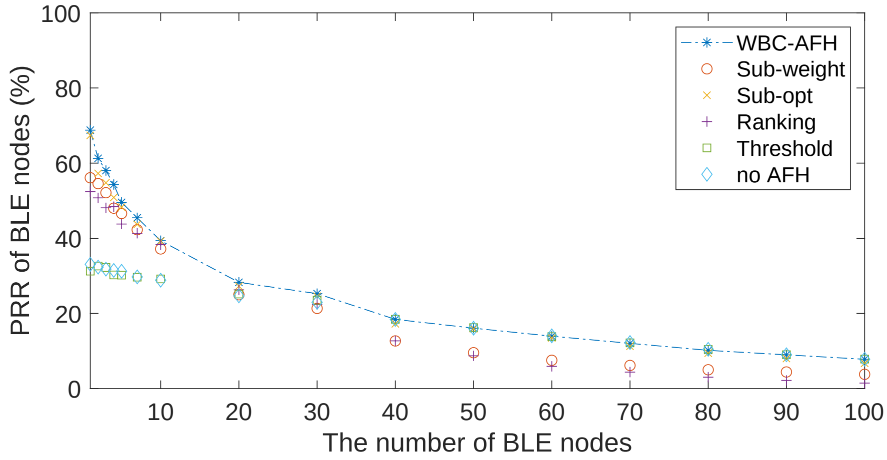


Figure 3.10: Packet reception ratio with CI=10 msec, 250 kbps.

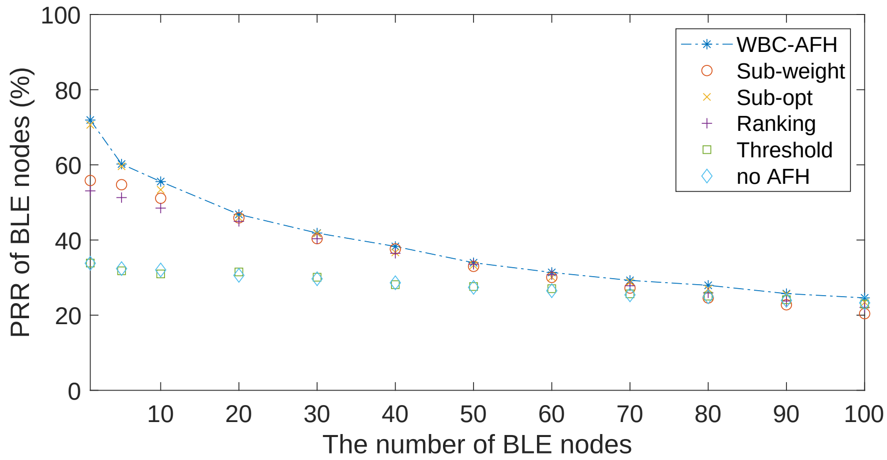


Figure 3.11: Packet reception ratio with CI=40 msec, 60 kbps.

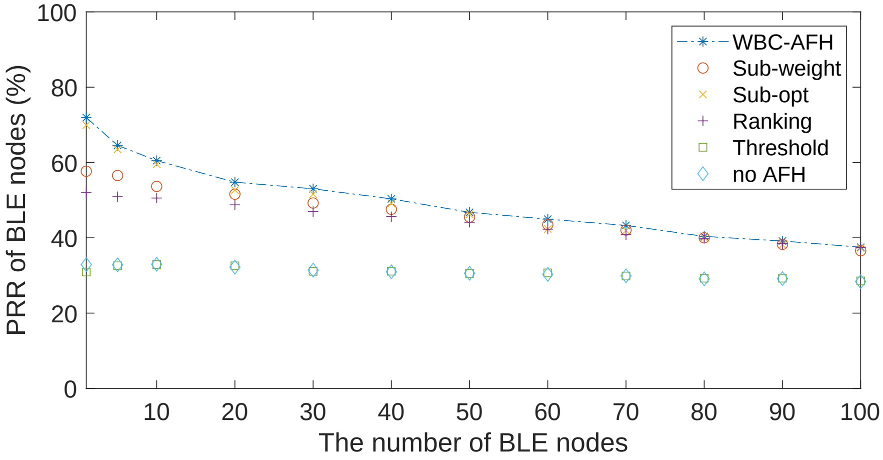


Figure 3.12: Packet reception ratio with CI=100 msec, 25 kbps.

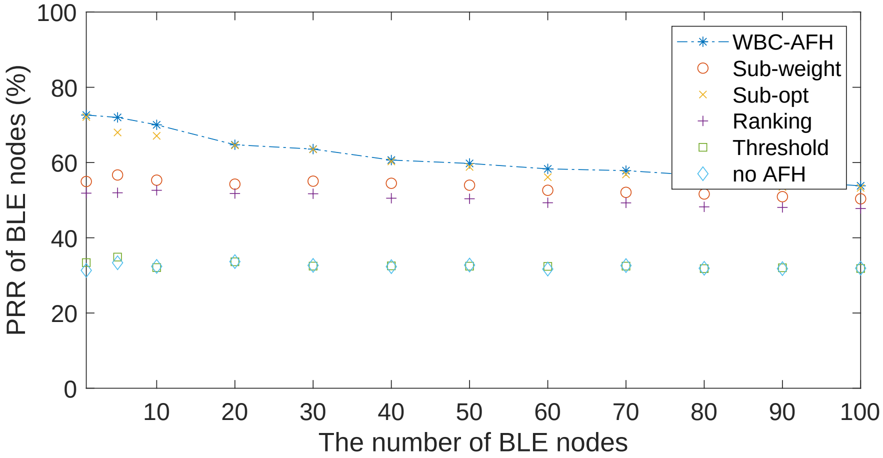


Figure 3.13: Packet reception ratio with CI=400 msec, 6 kbps.

small number of nodes or a low traffic rate. However, these two methods show lower performance than threshold-based in a situation where BLE traffic is dense. These results mean that the parameters of these techniques should be properly tuned according to the environment. These methods require appropriate parameter tuning whenever the channel environment changes, making it difficult to adapt to a dynamic environment. We can point out that ranking-based AFH also has limitations. Unlike the above methods, the proposed method can overcome limitations because it automatically adjusts the parameters according to the surrounding environment.

The hopping set size optimization, a submodule of the proposed method, also shows good performance. This result means that solving the optimization problem using CTC information greatly helps in improving collision in a congested band. Nevertheless, it shows 1% to 4% lower performance than the proposed method in a high data traffic environment. Therefore it needs the help of the submodule of channel weight.

3.5 Future Work

Although this chapter addresses AFH of BLE, we can apply a similar approach to other wireless communication.

In the case of Bluetooth classic, most operations are similar to BLE, and the number of channels is different from BLE due to the difference in bandwidth. However, we can apply the same method since the primary AFH mechanism is the same.

In the case of the ZigBee network, the frequency hopping-based TSCH has a similar aspect to the BLE system. The timeslot of TSCH can correspond to the connection interval of BLE, and direct CTC information delivery through Wi-Fi also corresponds to CTC techniques such as WeBee. However, there are several challenges to applying *WBC-AFH* to ZigBee. First, unlike BLE, which forms a small piconet between master and slave, TSCH generally constitutes a multi-hop network. Therefore, the external interference experienced by each node in the network may be different. Second, the channel map update transmitted directly from the master to the slave must be transmitted network-wide in TSCH. This situation causes a negotiation situation, which causes additional issues. As a prototype, we will check whether *WBC-AFH* can be applied to TSCH in a small multi-hop network situation and find a way to solve the abovementioned challenges.

3.6 Summary

We addressed the frequency hopping issues of BLE in congested ISM band. We design the *WBC-AFH* system that the BLE node exploits the information from Wi-Fi network with CTC and selects frequency hopping set with weighted manner. We mathematically derive an optimal size of channel set and weight of each BLE channel for a given channel condition. We evaluate the performance of the *WBC-AFH* both through implementation based experiment and extensive simulations. The results show that significant performance improvement comparing to conventional AFH.

Chapter 4

Conclusion

In this dissertation, we address two approaches to overcome IoT's limitation.

First, we have proposed a novel layered architecture using Wi-Fi and BLE. We have designed *Wi-BLE* to maintain a network with low power consumption and also provide high data rate occasionally. Finally we have evaluated the performance of *Wi-BLE* through extensive experiments in an indoor testbed. We have compared the performance results with that of Wi-Fi network using AODV routing protocol and the results show that our proposed *Wi-BLE* significantly reduces the energy consumption of nodes in our testbed. With *Wi-BLE*, IoT networks can achieve a wider range and high data rate with low energy.

Secondly, we have addressed the frequency hopping issues of BLE in congested ISM band. We have designed the *WBC-AFH* system that the BLE node exploits the information from Wi-Fi network with CTC and selects frequency hopping set with weighted manner. *WBC-AFH* mathematically derives an optimal size of channel set and weight of each BLE channel for a given channel condition. Finally we have evaluated the performance of the *WBC-AFH* both through implementation based experiment and extensive simulations and the results show that significant performance improvement comparing to conventional AFH. With *WBC-AFH*, IoT networks can achieve better reliability in extremely congested bands.

To summarize, we broke through the seemingly impossible requirement through cooperation between heterogeneous IoT devices. By solving the challenges of newly required IoT applications, IoT applications will become more diversified and advanced. Although some issues remain to be addressed, we believe IoT still has significant potential to provide a wider variety of applications we could not imagine. We hope this dissertation can become a touchstone for the richness of IoT.

Bibliography

[1] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, *Ad hoc On-Demand Distance Vector (AODV) Routing*, RFC 3561, Jul. 2003. DOI: 10.17487/RFC3561. [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. Networks*, 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 *Second International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS 2008)*, 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 *2006 IEEE 63rd Vehicular Technology Conference*, IEEE, vol. 1, 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 *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, Citeseer, 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 Journal of Computer and Information 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 *2015 3rd International Conference on Future Internet of Things and Cloud*, IEEE, 2015, pp. 558–563.

[11] B. Al Nahas, S. Duquennoy, and O. Landsiedel, “Concurrent transmissions for multi-hop bluetooth 5.,” in *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 *2014 6th European Embedded Design in Education and Research Conference (EDERC)*, IEEE, 2014, pp. 172–176.

[13] T. Lee, M.-S. Lee, H.-S. Kim, and S. Bahk, “A synergistic architecture for rpl over ble,” in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, IEEE, 2016, pp. 1–9.

[14] K. Mikhaylov and J. Tervonen, “Multihop data transfer service for bluetooth low energy,” in *2013 13th international Conference on ITS Telecommunications (ITST)*, IEEE, 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 Communications and Mobile Computing*, vol. 2017, 2017.

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

[18] T. Clausen, P. Jacquet, C. Adjih, *et al.*, "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 networking*, 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 Transactions on Networking*, 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 and Microsystems*, 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 *2016 Euromicro conference on digital system design (DSD)*, IEEE, 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 *2015 IEEE conference on computer communications (INFOCOM)*, IEEE, 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 *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2016, pp. 1–6.

[27] D.-J. Deng, M. Gan, Y.-C. Guo, *et al.*, "Ieee 802.11 ba: Low-power wake-up radio for green iot," *IEEE Communications Magazine*, 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 *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, IEEE, 2016, pp. 1–9.

[29] W. Park, D. Ryoo, C. Joo, and S. Bahk, "Bless: Ble-aided swift wi-fi scanning in multi-protocol iot networks," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, IEEE, 2021, pp. 1–10.

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

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

[32] H. Qin and W. Zhang, "Zigbee-assisted power saving for more efficient and sustainable ad hoc networks," *IEEE transactions on wireless communications*, vol. 12, no. 12, pp. 6180–6193, 2013.

[33] S. Bluetooth, “Bluetooth 3.0,” *HS specification, vol 0*, vol. 21, 2009.

[34] 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 Transactions on Sensor Networks (TOSN)*, vol. 15, no. 4, pp. 1–25, 2019.

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

[36] M. Baddeley, A. Aijaz, U. Raza, *et al.*, “6tisch++ with bluetooth 5 and concurrent transmissions,” in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, 2021, pp. 25–30.

[37] H.-S. Kim, J. Paek, and S. Bahk, “Rplie: Rpl for indoor environments under midterm link fluctuations,” *Journal of Communications and Networks*, vol. 23, no. 3, pp. 201–211, 2021.

[38] B. SIG, *Core specification 5.3*, 2017.

[39] M. Spörk, J. Classen, C. A. Boano, M. Hollick, and K. Römer, “Improving the reliability of bluetooth low energy connections.,” in *EWSN*, 2020, pp. 144–155.

[40] V. Poirot and O. Landsiedel, “Eafh: Informed exploration for adaptive frequency hopping in bluetooth low energy,” *arXiv preprint arXiv:2112.03046*, 2021.

[41] K. Chebrolu and A. Dhekne, “Esense: Energy sensing-based cross-technology communication,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2303–2316, 2012.

[42] D. Croce, N. Galioto, D. Garlisi, G. Giaconia, F. Giuliano, and I. Tinnirello, “Busybee: Low rate wifi-zigbee communications without gateways,” in *WiN-TECH’14*, 2014.

[43] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 317–330.

[44] Y. Zhu and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*, 2005, pp. 140–147.

[45] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "Zifi: Wireless lan discovery via zigbee interference signatures," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, 2010, pp. 49–60.

[46] Z. Chi, Y. Li, H. Sun, Y. Yao, Z. Lu, and T. Zhu, "B2w2: N-way concurrent communication for iot devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 2016, pp. 245–258.

[47] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 2–14.

[48] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He, "Bluebee: A 10,000 x faster cross-technology communication via phy emulation," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–13.

[49] W. Jiang, S. M. Kim, Z. Li, and T. He, "Achieving receiver-side cross-technology communication with cross-decoding," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 639–652.

[50] Z. Li and T. He, "Longbee: Enabling long-range cross-technology communication," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 162–170.

[51] Y. Chen, Z. Li, and T. He, "Twinbee: Reliable physical-layer cross-technology communication with symbol-level coding," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, 2018, pp. 153–161.

[52] X. Guo, Y. He, J. Zhang, and H. Jiang, "Wide: Physical-level ctc via digital emulation," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, IEEE, 2019, pp. 49–60.

[53] X. Guo, Y. He, X. Zheng, Z. Yu, and Y. Liu, "Lego-fi: Transmitter-transparent ctc with cross-demapping," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6665–6676, 2021.

[54] S. Wang, S. M. Kim, and T. He, "Symbol-level cross-technology communication via payload encoding," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2018, pp. 500–510.

[55] Z. Yin, Z. Li, S. M. Kim, and T. He, "Explicit channel coordination via cross-technology communication," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 178–190.

[56] L. Li, Y. Chen, and Z. Li, "Wible: Physical-layer cross-technology communication with symbol transition mapping," in *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, IEEE, 2021, pp. 1–9.

초 록

사물인터넷은 현재 다양한 영역에서 application을 개척하여 생활화되어 왔다. 이 학위 논문에서는 사물인터넷의 응용 사례 확장을 위해 에너지 효율적인 전송 속도 향상, 저전력 상황에서의 전송 범위 확장, 혼잡한 대역에서의 신뢰성 향상을 새로운 도전 과제로 삼고, 이러한 도전 과제를 극복할 두 가지 주제를 다룬다.

첫째, 다중 홉 네트워크 상황에서의 블루투스 저전력과 Wi-Fi 네트워크를 결합한 새로운 구조의 계층적 네트워크 시스템을 설계하고 이에 기반한 에너지 효율적인 전송 속도 향상 및 저전력 상황에서의 전송 범위확장을 제안한다. 제안된 시스템은 Linux 커널에 구현하여 실내 테스트베드를 통해 성능을 평가한다. 결과적으로 제안한 기법이 다중 홉 상황에서 기존 블루투스 저전력/Wi-Fi 단일 ad-hoc 네트워크와 비교하여 높은 데이터 트래픽을 지원하며, 테스트베드에서의 평균 전력 소비를 줄이는 것을 확인한다.

둘째, Cross-technology Communication (CTC)과 채널 가중치를 통한 블루투스 저전력의 Adaptive Frequency Hopping (AFH) 문제를 해결한다. 최종적으로 블루투스 저전력 기기가 사용하는 채널에 가중치를 두는 AFH 기법을 설계하여 Wi-Fi와 블루투스 저전력 기기가 모두 혼잡한 대역에서의 신뢰성을 향상한다. 프로토타입 실험과 시뮬레이션을 통해 제안한 기법이 기존의 AFH 기법과 비교하여 혼잡한 ISM 대역에서 블루투스 저전력의 패킷 수신율을 증가시키는 것을 확인한다.

주요어: 사물인터넷, 멀티홉 네트워크, Wi-Fi, 저전력 블루투스, 주파수 호핑
학번: 2014-21743