Radio Diversity for Heterogeneous Communication with Wireless Sensors

Yuan Qin  
Electrical and Electronic Engineering  
Imperial College London  
London, UK  
yuan.qin14@imperial.ac.uk

David Boyle  
Dyson School of Design Engineering  
Imperial College London  
London, UK  
david.boyle@imperial.ac.uk

Eric Yeatman  
Electrical and Electronic Engineering  
Imperial College London  
London, UK  
e.yeatman@imperial.ac.uk

Abstract—Unmanned Aerial Vehicle (UAV) deployed in tandem with wireless sensor networks (WSN) are increasingly considered for remote sensing applications. To date, most reported implementations continue to rely on communication protocols designed for static WSN that often require synchronization. Synchronous protocols, such as ZigBee, have long association times before the nodes synchronize, which compromises performance for aerial data collection. Asynchronous protocols based on the IEEE 802.15.4 physical layer tailored for mobile WSN data collection have been proposed. However, low over the air bit rates limit the amount of sensing data transferred for each UAV visit. To obtain a higher bit rate, we have developed a dynamic protocol capable of switching between physical layers of IEEE 802.15.4 (IEEE mode) and Bluetooth Low Energy (BLE mode), which works as an extension of UAV Integrated WSN Protocol (UIWP). The BLE mode fits into the UIWP Data State which manages burst transmissions of data. BLE mode is implemented at a low cost in terms of RAM and ROM usage, which are several hundreds of bytes respectively. Experimental evaluation results show that BLE mode consumes 54% of time and 57% of energy for transmitting the same amount of data compared with IEEE mode.


I. INTRODUCTION

Unmanned aerial vehicle (UAV), operating as a mobile agent, is adopted as a key practical entity for the wireless sensor network (WSN) research. Comparing to other terrestrial mobile agents, UAVs are less constrained with regard to the maneuverability, operation range, and mobility. The advantages of UAV attract researchers’ interests in developing new sensing applications [1]–[3], modeling the mobility pattern for the data collection [4], [5], revising the existing protocols for mobility needs [6], [7], or designing a bespoke communication protocol [8], [9].

As suitable communication protocols for such sensing applications are still in the early stage of research and development, mobility pattern modeling is less practically investigated. Applications incorporating UAV, or mobility pattern modeling, assume well developed low energy communication protocol for static WSN, such as IEEE 802.15.4e TSCH, Bluetooth Low Energy (BLE), and Zigbee based on IEEE 802.15.4. The aforementioned protocols are usually energy optimized for static WSNs using a synchronization process. This approach is enabled in a static WSN leveraging known fixed locations of nodes. Using mobile data collection approach makes it difficult to schedule and implement synchronization, or to be synchronized when not in the vicinity of other nodes, or when speed of flight limits proximity time between sink and node. The deficiencies due to mobility make them less attractive with respect to energy consumption and latency performance.

To solve this issue, Qin et al. proposed an asynchronous communication protocol, namely the balanced network communication protocol, with the assumption that sensor nodes in the network are homogeneous [8]. They further extended the work and proposed UAV integrated WSN protocol (UIWP) for a generic wireless sensing system with aerial data collection [10]. IEEE 802.15.4 physical layer (IEEE mode) in 2.4 GHz ISM band is adopted for the protocol, which limits the bit rate to 250 kbps. However, for sensing applications with high data volumes, such as triggered image recording, using a low bit rate radio is less likely to meet the application needs, given the short access times of a mobile sink.

BLE physical layer (BLE mode) has an over the air bit rate of up to 2 Mbps, which significantly boosts the amount of data that can be transferred within a specific time interval. However, the standard BLE protocol implementation carries out a synchronization process before connection state data sharing. A possible approach is to naively send data in BLE physical layer when necessary. Higher layer protocols are needed to manage switching to BLE mode.

This paper extends the work presented by Qin et al. in [10]. We implement online physical layer protocol switching between IEEE mode and BLE mode for the multi-protocol supported system on chip (SoC). The BLE mode would work within UIWP framework when certain conditions are met (discussed in section III-A). Data rate boosting in BLE mode only happens in the UIWP Data State. To the best of our knowledge, this is the first work to discuss the use of two physical layers in the design of a protocol for mobile aerial data collection from wireless sensors. The main contributions of this paper are twofold:

- Implementation of online IEEE mode and BLE mode.
IEEE 802.15.4 physical layer based protocols are explored in protocol design for hybrid mobile wireless sensing applications. MTSCH protocol proposed by Al-Nidawi and Kemp aims to adapt for mobility [6]. By properly tuning the association related parameters, the association process is facilitated. The latency is minimized by leaving the disassociated. Qin et al. designed the balance network communication protocol for a mobile aerial sink to reliably collect sensed data [8]. The throughput for for a number of realistic conditions were experimentally evaluated. Valente et al. constructed a WSN system with an aerial robot for vineyards health condition monitoring [1]. The system proves that a mobile node is capable of carrying out remote sensing missions. The use of IEEE 802.15.4 physical layer, however, is applicable only for applications with low data volume requirement. High data volume of a sensor node extends the UAV hovering time and reduces the system’s overall efficiency.

In contrast, BLE attracts researchers attention due to its low energy profile, compatibility with the Bluetooth protocol, as well as high bit rates. Spork et al. proposed BLEach that supports IPv6, which claimed to be the first open-source stack with complete support for IPv6 over BLE [11]. BLEach is integrated into the Contiki operating system [12] with the concept of rime stack [13]. It is demonstrated to be lightweight, energy efficiency, and interoperable with other BLE devices. A seamless BLE connection migration that is mobility aware is achieved by SeamBlue [14]. However, both BLEach and SeamBlue are standards compliant protocols that require synchronization before data sharing. The latency incurred prior to synchronization affects the performance of the mobile sink significantly. The Contiki community implements BLE advertising process in addition to the rime stack that supports IEEE 802.15.4 stack in Contiki OS [15]. This implementation only advertises beacons in 3 BLE advertising channels. Advertising, reception and communication are not realized.

Similar functionality and profile between IEEE 802.15.4 and BLE led to a variety of comparative analyses aiming to determine which protocol outperforms the other in different situations. The energy consumption measurement conducted by Siekkinen et al. shows that BLE is more energy efficient in terms of number of bytes transferred per Joule, which consumes energy 2.5 times less than that of IEEE 802.15.4 [16]. Mikhaylov et al. presented a performance analysis among BLE, IEEE 802.15.4, and SimplicityTI [17]. Results showed that application layer throughput with full BLE stack underperforms other protocols, while BLE only requires 2 to 7 times less energy to transfer data. Comparative analysis results show that high bit rate of BLE physical layer improves the net energy efficiency.

III. DESIGN AND IMPLEMENTATION

We implement online IEEE and BLE mode switching for UIWP in Contiki OS [12]. The BLE mode is designed for naive data bursting. Other processes, such as sensor node discovery and data collection scheduling, are handled by the UIWP in IEEE mode. We firstly give an overview of UIWP and discuss under which situations BLE mode would be enabled. Then the timing of BLE mode switching is presented. We finally illustrate how the BLE mode is operated within the UIWP framework and Contiki OS.

A. Overview of UIWP

UIWP is a protocol designed for a variety of wireless sensing applications with mobile data collection and device interactions, primarily taking wireless charging capabilities into account. The protocol is based on the 2.4 GHz ISM IEEE 802.15.4 physical layer. UIWP-MAC is built upon physical layer, which is an asynchronous hybrid medium access control (MAC) protocol. UIWP-MAC runs ContikiMAC on sensor nodes to perform duty cycling, while keeping the radio always on at the sink. UIWP-MAC for the sink sends a packet multiple times for each starting packet to fulfill the ContikiMAC channel check timing requirements. UIWP-APP is an application layer protocol that lies above UIWP-MAC. It defines the rules on how the sink discovers sensor nodes, collects sensor nodes’ information, implements the algorithm to prioritize sensor nodes, and collects sensed data. UIWP-APP for the sink node has four states; namely Advertise State, UIWP Ack State, Request State, and Data State, as shown in Figure 1. UIWP-APP for the sensor node

1https://github.com/yqinic/yqinic.github.io

![UIWP states diagrams.](Image)
has Listening State and Sending State that passively respond to the sink node’s commands.

Except for the UIWP Data State, only several packets are transferred within the network. For this reason, BLE mode implementation is predominantly limited to the UIWP Data State. UIWP Ack State packet format defines the size of sensed data to be transferred. The sink node would make a decision on whether to switch to BLE mode for a specific sensor node upon the reception of UIWP Ack State packet. Switching should be scheduled if data volume exceeds threshold $T_{dls}$. However, the calculation of $T_{dls}$ should be application dependent, and is out of the scope of this paper. After the decision has been made, the sink node notifies the sensor node which mode and which channel they are going to communicate on via the Request State packet.

### B. Timing of BLE Mode Switching

BLE mode works predominately throughout the UIWP Data State, if the sink node decides to use BLE mode for sensor data reception. It also works on UIWP Request State to wait for data packets come in after Request packet frame is sent. For sensor nodes, BLE mode only works on Sending State. The UIWP states that BLE mode works during are indicated in Figure 1.

Upon successful reception of IEEE 802.15.4 physical layer acknowledgment in UIWP Request State, the sink node switches to the BLE mode, as shown in Figure 2. Expiry timers $t_{dcs}$ and $t_{adc}$ defined in [10] are applied here for the sink node. IEEE mode is switched back if either $t_{dcs}$ or $t_{adc}$ expire. For the sensor node, protocol switches to BLE mode if Request State data frame, which contains the sink node’s notification to work on BLE mode, is received. IEEE mode is switched back if either $t_{adc}$ expire or data transferring complete. The mode switching schedules for the sink node and the sensor node are illustrated in Figure 2.

### C. Design Concepts and Implementation

This work, as a BLE mode extension for UIWP, is implemented in Contiki OS and TI CC2650 SoC, which is a multi-protocol SoC that supporting both IEEE 802.15.4 and BLE 4.2 specification. There are two possible ways of BLE mode implementation: full stack approach and function approach. For the full stack approach, full rime stack that supports BLE, such as BLEach [11], would be programmed in addition to IEEE 802.15.4 stacks. Protocol switching is achieved by calling different stack drivers. Unnecessary drivers may be switched off.

The BLE mode is aimed to be utilized for naive data bursting, which requires only basic functionalities such as sending, listening, and packet handling. The full stack approach is too heavy that would have the same level of random access memory (RAM) and read only memory (ROM) usage as IEEE 802.15.4 stacks has, yet synchronization and duty cycle scheduling are not required for BLE mode. Hence the function approach, which only carries out required functionalities, is optimal.

The Contiki community programs the BLE mode advertising by switching off the IEEE mode temporarily. The IEEE mode is restored, after advertising process is accomplished [15]. We borrow this concept in our work.

The sensor node BLE mode switching is completed by blocking the UIWP-MAC duty cycled channel check and other IEEE mode radio functionalities. BLE mode for the sensor node works in Sending State. It is in charge of the sending process. The sensor node also supports data management that chops the data into short packets for BLE mode, as CC2650 radio working on BLE mode supports up to 31 bytes per protocol data unit [18]. Sensor data is sent continuously, until the Sending State to Listening State transition conditions are met. BLE mode is then switched off and IEEE mode is restored.

For the sink node, BLE mode carries out listening and packet parsing processes. Similarly, IEEE mode functionalities are blocked while BLE mode is in operation. During the BLE mode listening, interrupt handler is enabled to handle the incoming BLE packets. Packet parsing function is executed whenever the incoming BLE packets are available. IEEE mode is restored when the Data State to Advertise State transition conditions are met.

### IV. Evaluation

To evaluate the performance in BLE mode, we devise a case study that applies the UIWP and BLE mode switching to a wireless sensor application with aerial data collection that equips camera traps for a wildlife monitoring study, where a UAV is used for data collection from the camera trap sensors.
A. Sensing Data Size Modeling

The animal community monitoring presented by Kays et al. deployed a maximum of 20 digital camera traps for a period of 1 year [19]. The experiment was conducted at Barro Colorado Island, Panama. Each camera was deployed in a location for a consecutive 8 days, and shifted to the next. The study resulted in totally 764 deployments with 17111 detection of animals. A daily average of 2.8 animals detected can be derived from each deployment location. We assume this number as the times that camera traps triggered daily in our modeling.

To evaluate the BLE mode performance, we focus on the data transfer rate, which is the amount of data transferred per second in UIWP Data State, on a single camera trap sensor node. The assumptions made are summarized in Table I. Kay et al. suggested that 1 megapixel resolution would be sufficient for monitoring. We adopt this assumption. We assume 3 consecutive pictures are taken for each detection, and a 100 KB storage space is occupied for each picture. These assumptions result in a daily data size of 840 kB. If the UAV visits each camera trap daily, this is the amount of sensing data that would be collected by the UAV carried mobile sink. Ideally the data collection process would be completed within a few seconds with BLE’s over the air bit rate, or tens of seconds with IEEE 802.15.4. Due to the effects from layers above physical, the actual time for data collection should be experimentally tested.

B. Memory Footprint

We quantify the BLE mode memory usage in addition to UIWP in terms of RAM and ROM, before the experiment is carried out. Table II shows the RAM usage and ROM usage for UIWP sink node and UIWP sensor node respectively.

UIWP sink requires more RAM compared to UIWP sensor due to an extended buffer queue for a burst incoming packets handling. UIWP sensor only receives acknowledgment packet, so a relatively short buffer queue is sufficient for the reception process. Additional ROM usage for UIWP sink manages listening and packet parsing processes. Comparing to the UIWP sink, UIWP sensor does not have the packet handling requirement. A resource constrained UIWP sensor would further shorten or even remove the BLE buffer to save RAM.

C. Energy Usage Measurements

We present a comparative study between BLE mode and IEEE mode in terms of energy usage during UIWP Data State. As discussed in Section IV-A, 840 KB data is assumed to be collected for a daily visiting UAV. The energy consumption for sending this amount of data is tested. The size for each packet frame is set to 96 bytes in IEEE mode, resulting in a total transmission of 8,960 packets over the air. Due to the BLE protocol constraints, 31 bytes of data is transmitted for each BLE mode packet. A sum of 27,748 packets was to be transferred in total.

We used an oscilloscope to monitor the voltage drop over a 33 ohm resistor. The resistor was connected in series with TI CC2650 Launchpad. The Launchpad ran UIWP with only IEEE mode and UIWP with BLE mode for two tests. To have a fair comparison, a transmission power of 5 dBm is set to both modes. A 3 volt direct current power source that connects to both the resistor and the Launchpad closes the circuit.

The oscilloscope recorded the voltage reading during the data collection process in UIWP Data State for 100 seconds with a resolution of 95.38 microseconds. Both modes has a maximum transmission current of 11.07 mA. To calculate the energy usage, the voltage drop on the resistor is deducted from the 3 volt source. From the testing results, we derive an energy usage of 1.71 Joule for UIWP with only IEEE mode. There are 4,232 packets delivered during 100 seconds. The energy usage for UIWP with BLE mode consumes 1.83 Joule during the recording period. However, 24,760 packets are received.

We believe that 100 seconds recording period is sufficiently long to obtain an unbiased estimate of the average power consumed per packet sent. From this estimation, the total energy consumption is calculated and shown in Table III. The results indicate that BLE mode consumes 1.8 times less energy comparing to IEEE mode for transmitting the same amount of data, which is significant in resource constrained devices for wireless sensor applications.

D. Outdoor Experiment Setup

The outdoor radio environment differs significantly from indoors. To study the performance differences, we setup an outdoor experiment and compared it with our indoor measurements. The outdoor experiment was conducted in Richmond Park, London. A sensor node (TI CC2650 Launchpad) with 840 KB pseudo camera captured picture data is placed on ground to simulate the camera trap sensor. A mobile sink node (TI CC2650 Launchpad) connected to UAV (DJI Matrice 100 quadcopter) collected the data from the sensor node.

<table>
<thead>
<tr>
<th>Items</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average no. of animal detection</td>
<td>2.8 times per day</td>
</tr>
<tr>
<td>Pictures took for each detection</td>
<td>3 pictures</td>
</tr>
<tr>
<td>Picture resolution</td>
<td>1 megapixel</td>
</tr>
<tr>
<td>Storage size</td>
<td>100 KB per picture</td>
</tr>
<tr>
<td>UAV visiting frequency</td>
<td>1 time per sensor per day</td>
</tr>
<tr>
<td>Average daily data size</td>
<td>840 KB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>RAM (Bytes)</th>
<th>ROM (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIWP sink</td>
<td>344</td>
<td>692</td>
</tr>
<tr>
<td>UIWP sensor</td>
<td>288</td>
<td>396</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Energy in Joule</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE Mode</td>
<td>2.05</td>
</tr>
<tr>
<td>IEEE Mode</td>
<td>3.62</td>
</tr>
</tbody>
</table>
UAV navigated to the sensor node according to the stored GPS coordinates. It will hover if it is notified that there is still data to be transferred. Then UAV flies to the next GPS coordinate after the data collection is completed. UAV keeps an altitude of 10 meters all the time.

The UAV was controlled by an onboard computer (we use Raspberry Pi 2 B+ for this work) over an UART-CAN2 cable. The sink node connects to the onboard computer over USB interface. A software serial interface written on the onboard computer enables it react on the sink node’s notification. In this experiment, the sink node will certainly raise a flag to notify the onboard computer to issue hover command, due to the stored data size. The sink node then notifies the onboard computer when the transmission finishes. UAV, onboard computer, and the sink node connections are shown in Figure 3.

During the experiment, the onboard computer records the amount of time the UAV hovers to wait for the data transmission to finish, which is also the time elapsed from the beginning of the sensor data reception to the data transmission completion. Similarly, there are two measurements: UIWP with only IEEE mode and UIWP with BLE mode. The results are discussed in the next section.

### E. Results and Discussion

The number of packets transmitted during 100 seconds were obtained from the energy usage measurements in the indoor environment. The time elapse for transmitting 840 KB camera trap sensing data is then scaled and plotted in Figure 4. The transmission in IEEE mode takes over 200 seconds, which nearly doubles the transmission time for this mode.

We also tested the performance of IEEE mode and BLE mode in an outdoor environment, and the results are plotted in Figure 4. Similar to the indoor test, the BLE mode transmission takes nearly half of the time (113.82 seconds) compared to that of the IEEE mode (210.08 seconds). From the test results, we can observe that there is no material difference for the protocol working in indoor and outdoor environments, no matter if it is in IEEE mode or BLE mode.

The outdoor test shows a realistic data transfer rate of 59 kbps in BLE mode, which is about 6% of the bit rate over the air. A data transfer rate of 32 kbps is realized for the IEEE mode, which is about 13% of its bit rate over the air. This explains that our implementation of BLE mode only has about a half of the transmission time advantage, instead of a quarter that is implied by the bit rate over the air. This percentage difference could be caused by the protocol headers. The data bits have a less percentage of occupation of a packet frame for BLE mode, due to a longer protocol header. A more efficient way of BLE mode implementation could possibly contribute to the bit rate utilization.

### V. Conclusion and Future Works

We present a practical implementation of BLE physical layer for data bursting with an IEEE 802.15.4 physical layer based UAV Integrated WSN Protocol. We discussed the timing and conditions to shift to BLE mode, and how the BLE mode is fitted to the UIWP framework. The memory footprint, energy usage, and the time elapse for transmitting a specific amount of data are evaluated for a case study. Comparing with UIWP with only IEEE mode, UIWP with BLE mode consumes 54% of time and 57% of energy.

Our BLE mode implementation is based on an asynchronous advertising process, which cannot share advantages of synchronization. Channel hopping, for example, is one of the advantages of synchronization, which effectively handles the packet collisions, interference, and channel fading. As discussed in the paper, the synchronization process follows a probabilistic node discovery process, which significantly drops the amount of data transferred within limited access time.

Given that sensor nodes are already discovered during UIWP Ack State working in IEEE 802.15.4 physical layer, synchronization could be established immediately in BLE mode. If an asynchronous protocol working on IEEE 802.15.4
physical layer immediately gets synchronized when it shifts to BLE physical layer, the data collection process for mobile aerial data collection from wireless sensors may be more reliable.

ACKNOWLEDGMENT

This work is supported by the China Scholarship Council (CSC) and the Department of Electrical and Electronic Engineering, Imperial College London. The authors also thank Dr. Hailing Fu for providing laboratory and equipment access.

REFERENCES


