Efficient and Reliable Aerial Communication with Wireless Sensors

Yuan Qin, *Student Member, IEEE,* David Boyle, *Member, IEEE,* and Eric Yeatman, *Fellow, IEEE*

***Abstract*—The abstract goes here. We'll write this last.**

***Index Terms*—Unmanned Aerial Vehicles, Wireless Sensor Networks, Communications Protocols, Data Collection, Cyber-physical Systems, Internet of Things.**

I. Introduction & Motivation

THIS paper extends work on developing communications between unmanned aerial vehicles and wireless sensors presented in [1], and contributes to the development of hybrid data and wireless power transfer systems [2]. Sensing systems incorporating unmanned aerial vehicles (UAVs) have the potential to enable a host of hitherto impractical monitoring applications exploiting wireless sensors in remote and extreme environments.

Using UAVs as mobile data collection agents can improve reliability by reducing or eradicating the need to route packets across a network towards a sink or gateway device. Typical wireless sensor networks (WSN) often fail to perform reliably in places characterized by difficult RF environments and treacherous maintenance conditions. The use of UAVs to wirelessly collect data from and deliver power to devices has the potential to overcome some of these challenges.

An immediate problem with using UAVs for wireless sensor data collection is that communications protocols have not been designed with the operational characterstics of such interactions in mind. There is little published literature concerning this problem, and no *off the shelf* solutions. The mobility of a UAV, particularly its speed, altitude and approach vector, introduces challenges for effective communications protocol design. This paper focuses on the design and evaluation of a novel communications protocol, UAV Integraetd WSN Protocol (UIWP), which takes into account the challenges presented by UAVs interacting with static devices under realistic, outdoor conditions. The main contributions of this paper include:

- a
- b
- c

The remainder of the paper is organized as follows:

II. Related Work

Research at the intersection of UAV and WSN technologies has attracted much recent interest. Valente *et al.* presented a wireless sensor and aerial robot system for crop monitoring in which aerial robots collect sensor data from several sparse clusters [3]. Costa *et al.* discuss a precision agriculture application using UAVs to spray chemicals on crops using sensor node feedback [4]. UAVs have also proved useful in place of satellite communications for marine environment monitoring [5]. UAVs combined with WSN are also reported for water management and distributed irrigation control [6], post-disaster monitoring [7], [8], and animal monitoring [9]. The structural health of UAVs can also be monitored by sensor nodes while carrying out a mission [10]. Traditional communication protocols for typical static WSN, e.g. ZigBee and LoRa, were used in these applications. For mobile aerial data collection, these protocols compromise desirable performance characteristics because of synchronization requirements.

Sensor data collection is the common motive for using UAVs with WSN applications. Many researchers have therefore focussed on the data collection process. Wang *et al.* model aerial data collection as a traveling salesman problem (TSP). They try to minimize the overall UAV path for data collection [11]. Olivieri and Endler made use of a dynamic set of UAVs for data collection, achieving 13% better efficiency over the TSP solution [12]. Particle swarm optimization to elect cluster heads is considered by Ho *et al.* [13], reporting that the UAV traveling time is reduced without sacrificing Bit Error Rate (BER) and energy consumption. To better understand UAV-based data collection, several models, strategies, and theoretical analyses are proposed in [14], [15], [16]. However, there are no practical results shown for these models or analyses. Rashed and Soyturk analyzed the effects of several UAV-specific factors, such as speed, altitude, and flight path, on the data collection process [17]. They introduced a new metric for selecting the appropriate mobility pattern, aiming to maximize the covered nodes and minimize operating time.

The majority of these studies provide little by way of practical evaluation and rely heavily on simulation and analytical analyses [11], [12], [13], [14], [15], [16], [17]. It has been found that traditional communications protocols for WSNs tend to perform poorly in practice [18], and researchers have begun to design bespoke protocols that specifically cater to aerial mobility. Qin *et al.* designed an application protocol for a mobile sink to reliably collect data with moderate UAV speeds [1]. Li *et al.* proposed A-OAloha for UAV-WSN system, intended to increase throughput while keeping acceptable BER [19]. Say *et al.* adjust the contention window value in IEEE 802.11 MAC to achieve priority selection, thus improving data collection efficiency [20]. Alshbatat and Dong adopted a cross-layer approach with Directional Optimized Link State Routing (DOLSR) protocol to tune key parameters to improve quality of service [21]. The decision for selecting

Manuscript received August xx, 2018

the route was based on information gathered from first three layers of the protocol stack, and they showed that the end-to-end delay is also minimized compared to IEEE 802.11 under simulation in OPNET. Cai *et al.* proposed a MAC scheme for ad hoc UAV networks to deal with full-duplex radios and multi-packet reception, designed to optimize for throughput and delay [22]. Kotsiou *et al.* proposed ME-ContikiMAC, based on ContikiMAC, to alleviate packet duplication and thereby reducing delay and energy consumption [23]. However, none of these protocol designs have carefully considered how mobility affects communication quality. A simple example is to consider the polarity of the sensor node antenna and how to determine the appropriate approach vector for a UAV.

IEEE 802.15.4e TSCH is one of the most promising recent MAC protocol designs for WSNs. The use of multiple channels increases reliability and network capacity. To adapt for mobility, Al-Nidawi and Kemp proposed the MTSCH scheme that focuses on reducing the latency incurred by the association and disassociation process [24]. They compared the impact of mobility between TSCH mode and low latency deterministic (LLDN) mode, described in [18]. The assumptions on mobility pattern and the probabilistic approach, however, constrain the usefulness of the approach.

III. UIWP Design

Communications between mobile aerial devices and static terrestrial devices introduce a number of interesting differences compared with typical WSNs. Taking speed, direction and altitude into account requires carefully evaluating the design and performance of existing communications protocols in parallel with considering new approaches. The following subsection details desirable characteristics of and design goals a protocol suitable for this type of system, prior to giving an overview of the design of UIWP and its implementation.

A. Design Goals

Characteristics of communications protocols for resource constrained devices typical of WSNs include *energy efficiency*, *reliability*, and *robustness*. For communication with mobile aerial devices that visit only occasionally, high data rates, low latency, channel diversity, and ability to prioritize devices' communications, are also likely to be important characteristics.

1) Energy Efficiency: The peak power consumption of radio transceivers typical of WSN-type devices (10s of mW for active IEEE 802.15.4, Bluetooth LE, etc., radios) is 3 to 4 orders of magnitude lower than that of a UAV (e.g. 300 W for a DJI Matrice 100 Quadcopter during flight). Therefore, minimizing the energy use of terrestrial nodes while ignoring that of aerial transceivers in the design of a suitable protocol is a rational approach.

2) Reliability: A mobile agent may associate or disassociate with a terrestrial sensor node at any time. A suitable protocol should reliably and efficiently detect such scenarios, and similarly handle the re-association process. Good fault tolerance, recovery, and data transfer reliability should also be characteristic of the protocol.

3) Robustness: The protocol should robust to collisions and contention where multiple sensor nodes may be simultaneously be in range of and attempting to communicate with the UAV.

4) High Data Rate: Applications employing UAVs will ideally minimize the amount of time during which the agent is proximate each terrestrial device for the purpose of data collection. Therefore, the rate of information exchange should be maximized, which in many cases requires data rate.

5) Low Latency: A low latency association process is a critical characteristic of a suitably designed protocol. Long association times reduce, if not prohibit, data transfer. Low latency of data transfer, facilitated by maximizing bandwidth, is also desirable to further minimize the total time of each interaction.

6) Channel Switching: Dynamic RF environments, particularly with mobility, require protocols to periodically assess channel quality and potentially switch to a more favorable one. Separating control and data channels is shown to be beneficial for improving link quality and avoiding hidden terminal problems under contention (Section III-C).

7) Priority: Given the limitations of UAV battery life and the fact that some wireless sensors may have higher criticality data to transmit or be in a more favorable RF environment, the ability to prioritize which devices interact with the UAV, and in what order, is desirable.

B. UIWP Architecture

This subsection describes the design of a novel protocol, UIWP, intended to satisfy the design considerations. UIWP is comprised of three layers: IEEE 802.15.4 PHY, UIWP-MAC, and UIWP-APP. The IEEE 802.15.4 PHY, which is available in many contemporary system on chip (SoC) intended for IoT applications, is adopted as the underlying physical layer. UIWP-MAC, a modified version of ContikiMAC [25], is built upon the physical layer. UIWP-APP is the application layer sitting above UIWP-MAC. UIWP-APP defines the rules by which the aerial agent discovers sensor nodes, collects sensor nodes' information, implements the algorithm to prioritize the sensor nodes, and defines how to transmit data. Channel selection and channel switching are implemented, as well as priority classification in UIWP-APP to deal with the mobility characteristics of aerial agents. The following subsections describe the design and implementation of UIWP-MAC and UIWP-APP.

1) UIWP-MAC: MAC layer protocols are easily divided into synchronous and asynchronous protocols. Synchronous protocols, such as IEEE 802.15.4e TSCH, make use of enhanced beacons (EB) for synchronization. Energy efficiency is achieved through properly scheduling the active timing of both transmitters and receivers.

The advantages of synchronization-based scheduling are difficult to exploit for mobile data collection from sensor nodes because of the association latency overheads, clock drift, and topology constraints. Communication happens only when sensor nodes are neighbors of the sink. Because of clock drift, the network can only exist for a short time period

without the synchronization. A better approach is to form an *ad-hoc network* when sensor nodes are neighbors of the sink. It is reported that it takes approximately 5 EB periods for a significant number of nodes (340) to be associated if 4 channels are employed [26]. The association process is thus energy and latency expensive for WSNs with mobile sinks relative to what may be achieved with an asynchronous approach. In addition to the association overhead, assuming a star topology (thus elimination of routing), the benefits of channel switching are limited as devices use one channel at a time.

In contrast, asynchronous MAC protocols allow radio duty-cycling for periodic listening or turn the radio on only if it has data to send; both of which significantly reduce energy consumed. Receiver, or sensor node initiated asynchronous MAC is implemented using Request to Send (RTS) followed by Clear to Send (CTS) mechanisms. A device ready to transmit data first send an RTS to a receiver. When a UAV-based receiver is available, it replies a CTS signal indicating that data delivery may begin [27]. This mechanism is not suitable for a mobile sink, as sensor nodes have no idea when the mobile sink will become its neighbor. Therefore, we focus on the sender-initiated MAC, under which the sink is in charge of notifying sensor nodes of its presence.

For static sensor nodes, we implement ContikiMAC [25]. At the UAV side, we keep the radio always on, sending each packet multiple times to guarantee that the packet is captured by sensor nodes. We call this hybrid approach UIWP-MAC. UIWP-MAC takes the following into consideration in its design: (1) Energy consumption of the static sensor nodes should be minimized. (2) Aerial nodes may be seen as equivalently mains powered. (3) Latency should be minimized. Since the mobile receiver is always on, a sensor node is required to send a packet one time for each interaction. An illustrative comparison between UIWP-MAC and ContikiMAC is shown in Figure 1. ContikiMAC checks the radio channel at a fixed channel check rate, which causes packet reception delay if radio is not yet switched on. If multiple packets are to be sent, the delay is accumulated and significantly decreases the effective data rate. By implementing UIWP-MAC, packet reception delay is eliminated at the sink side.

Reduction of receiver-side packet reception delay is important for mobile WSN. Due to the high chance of connection loss and the UAV's own energy constraints, improving the effective data rate is essential. UIWP-MAC takes advantage of always-on radio with dedicated one-to-one communication channels to maximize the effective data rate. Comparing to blockwise transfer and its implementation in Contiki [28], [29], UIWP-MAC does not experience the reception delay for the first packet. Furthermore, the block size for blockwise transfer is bounded by the queue buffer size. UIWP-MAC does not have this restriction so as to further boost the effective data rate.

2) UIWP-APP Overview: We use a state machine to describe how UIWP-APP works for UAVs collecting data from static sensor nodes. UIWP-APP has four states, namely *Advertise State*, *UIWP Ack State*, *Request State*, and *Data State*, shown in Figure 2. This is only used to describe the states

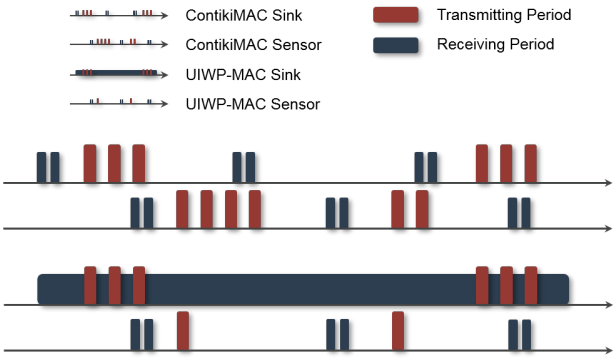

Fig. 1. Comparison of UIWP-MAC and ContikiMAC. Timelines (top to bottom): ContikiMAC sink, ContikiMAC sensor, UIWP-MAC sink, and UIWP-MAC sensor.

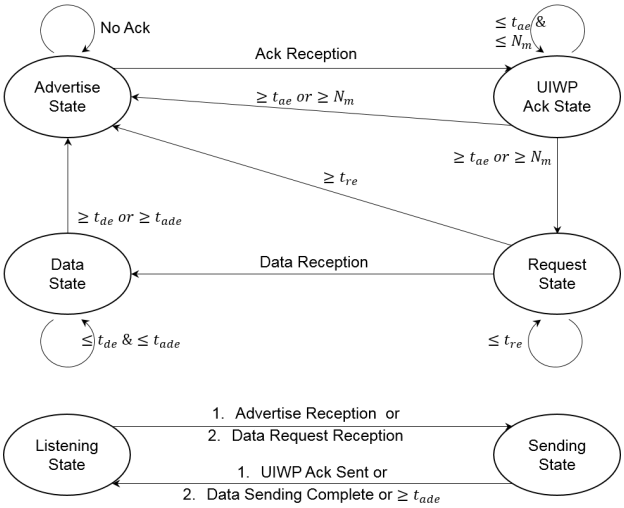

Fig. 2. Transitioning between states; UIWP sink (top) and UIWP sensor (bottom).

for the static sensor node. Devices respond passively to the aerial node's instructions. A delayed response may happen if connection loss is frequent as retransmission and backoffs loop continuously. A set of timers for the state machine are implemented to mitigate this problem.

The advertise state is the default. This starts and runs continuously when other states expire. In the advertise state, the aerial receiver broadcasts the purpose of the visit and priority class information after each broadcast interval t_b, i.e. time elapsed between the beginning of two consecutive broadcasts. Static sensors corresponding to both the purpose of a visit and the same level of priority (Section III-D) will respond. In this part of the session, the UAV actively searches for sensor nodes that meet the application requirements while corresponding neighboring nodes try to respond. The broadcast interval t_b is application specific. However, a small value is typically assigned to allow the discovery process to be as efficient as possible.

The UIWP ack state follows the advertise state when the first sensor node's response is captured by the receiver. Expire time t_{ae} is then set to indicate when this state will end. This application layer acknowledgment contains information

about the sensor node itself, such as battery information, GPS coordinates and priority information. A priority list is initiated at the start of this state. The acknowledged sensor nodes are fitted to the list according to a priority classification algorithm. UIWP ack state ends when either the t_{ae} expires or MAC_ACK_NUM is reached (defined maximum number of acknowledgments).

For applications not requesting sensor data, e.g. route planning, visual inspections, and wireless charging, UIWP returns to the advertise state when UIWP ack state ends; otherwise request state starts. In the request state, the receiver sets two expiry times. These are a specific sensor node's data request expiry time t_{ne} and the session expiry time t_{re}. The UAV requests data from the next device in the priority list if t_{ne} is reached. This state ends if the sink captures the data packet or t_{re} expires.

The sink node passively listens to the channel in the data state. Expiry timer t_{ade} is set to end the session if the data transmission cannot finish within this time period. Expiry timer t_{de} is also set if the receiver does not receive any data within this period.

C. Channel Scanning & Channel Switching

UIWP uses two different channels for communication to alleviate problems with collision, interference, and fading. A dynamically allocated data channel is used for the data state. The other three states use a predetermined fixed control channel. A fixed control channel is used such that the sensor node discovery process can be deterministic. A probabilistic discovery process, like IEEE 802.15.4e TSCH, that randomly selects a listening channel in each time slot, follows binomial distribution in observing new nodes. The latency may exceed the expected value significantly, it is possible that no nodes would be discovered even thought they are neighbors of the sink. Owing to the small number of packets exchanged (one packet for each of the UIWP Ack State and Request State) which may be more tolerant to RF channel quality variations, a fixed control channel is appropriate for the protocol.

To select an optimal data channel for current location, periodic Received Signal Strength Indicator (RSSI) measurement over all channels IEEE 802.15.4 2.4GHz ISM band is performed by the UAV's transceiver during the Advertise State. The sensor node passively receives the optimum channel measured by UAV.

Figure 3 demonstrates how channel switching operates for the sink and sensor nodes for each state. The data request frame sent by the sink contains the data channel information. The sink node switches to the data channel and waits for data when it receives the physical layer acknowledgment from the sensor node. The Data State starts once the data packet is detected, otherwise the sink node stays in the Request State if there is no packet seen within t_{ne}. If the sink is already in the Data State, the protocol checks whether t_{de} has been reached to determine if whether or not to stay in the Data State. The same process is followed for the remaining data packets.

The sensor node switches to the data channel once it has parsed the data request packet. Transmission of data packets

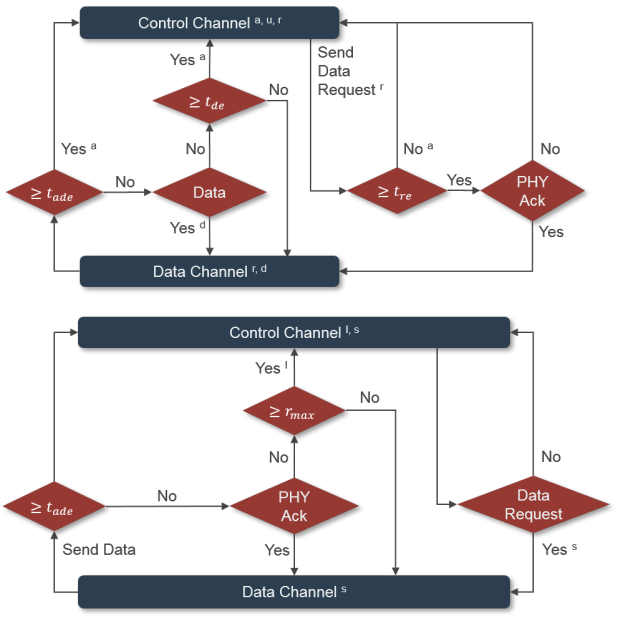

Fig. 3. Switching between Control and Data channels for UAV (top) and sensor nodes. Superscripts a, u, r, and d indicate Advertise State, UIWP Ack State, Request State, and Data State for the sink. Superscripts l and s are Listening and Sending states of the sensor node. Superscripts indicate current or next state.

takes place until t_{de}, or no physical layer acknowledgment is received within t_{ne}.

D. Priority & Classification

Wireless nodes may have sensors or associated tasks with different levels of importance in a sensing system. Also, given limited proximity time between a UAV and static sensors, it is intuitive that nodes with the best link quality should be afforded higher priority to communicate with the sink (i.e. they could be closer). Classification provides such mechanism to determine the order of sensor nodes for communication taking into account link quality and information priority.

The classification algorithm calculates the priority of sensor nodes, and writes them into a *priority list*. Figure 4 shows how the priority list updates and when it will be destroyed. A priority list is created at the end of acknowledgment session once t_{ae} expires or MAX_ACK_NUM is reached. The protocol sends a data request to the first sensor node in the list. If the data packet is received, it keep receiving data until the transmission is completed or the timer expires. The protocol determines whether to update the priority list based on if t_{re} has expired and n_l, which are the number of nodes in the list, has ran out, i.e. when there is no data received. If neither conditions is satisfied, the protocol updates the list. The device at the top of the list is then downgraded or deleted. A data request is then sent to the next node in device in the queue.

The classification algorithm sets the rules about how to create and update the priority list, and as such the algorithm tuned per application. A simple implementation is demonstrated in this paper for evaluation purposes. Static nodes are assigned a priority on deployment based on their importance to the application. The Advertise State embeds the priority class in

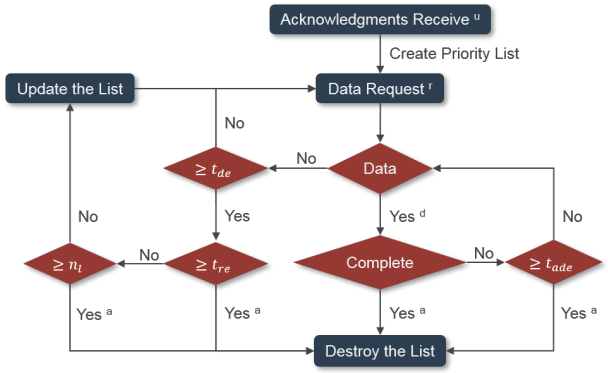

Fig. 4. Priority list update rule. Superscripts a, u, r, and d are the Advertise State, UIWP Ack State, Request State, and Data State for the sink node. Superscripts indicate current or next state.

the frame, such that only the demanded classes respond to the broadcast. For devices with the same class, priorities are determined by rank based on descending RSSI values.

A retransmission variable is attached to each device in the list. This limits the retransmission times of data request. In our implementation, we allow 3 retransmissions. The device is moved to the bottom of the current class if the first request fails. Then the device degrades to the next class, and the rank is based on RSSI if the second request fails. Finally, the device is deleted from the list if it remains unresponsive.

E. Integration with UAV

The UAV uses information from devices to perform actions including adjusting route and speed, determining whether to perform wireless charging or inspection, for example, in addition to storing and analyzing data. The UAV processor typically is passively controlled using an API, so we use an onboard computer programmed to interact between UAV and receiver.

Each mission is initiated and accomplished based on the programming of the onboard computer. The program starts by communicating with the UAV's systems and receiving radio to determine present location and put the mission in context. If the initiation process is completed without error, the program commands the UAV fly according to either relative positions or GPS coordinates. It uses the receiver's data (i.e. the interactions with static sensor nodes) to determine whether to reduce speed, maneuver or hover over a specific location. When the mission is complete, or the UAV battery approaches depletion, the UAV automatically returns to base.

F. UIWP Frame Format

There are four types of frame specified for UIWP. These are distinguished by first two bits at the beginning of the frame. They are *Advertise Frame*, *UIWP Ack Frame*, *Request Frame*, and *Data Frame*. These correspond to each state, and are encoded 00, 01, 10, and 11 respectively. The Advertise Frame and the Request Frame are sent by the sink node, while static devices send only the UIWP Ack Frame and Data Frame. Table I shows the detailed protocol formats.

TABLE I
PROTOCOL FORMAT

Advertise Frame				
Bits	0 - 1	2 - 5	6 - 12	
Functions	Frame Type	Mission	Priority	
UIWP Ack Frame				
Bytes	0	1 - 4	5 - 6	7 - 8
Functions	Type & Priority	Bytes	Voltage	Volume
Bytes	9	10 - 13		14 -
Functions	Antenna Type	Antenna Orientation		Camera
Request Frame				
Bytes	0			1 - 4
Bits	0 - 1	2 - 5	6 - 7	/
Functions	Frame Type	Channel	Order	Bytes
Data Subunit Frame				
Bits	0 - 3	4 - 31	32 - 63	
Functions	Data Type	Time Stamp	Data	

TABLE II
LIST OF MISSIONS AND ASSOCIATED SENSOR NODES

Value	**Missions**	**Associated Nodes**
0000	Check Nodes Presence	All
0001	Data Collection	Data Available
0010	Inspection	Application Specific
0011	Wireless Charging	Battery below Threshold
0100 -	Reserved	/

1) Advertise Frame: The Advertise Frame contains the mission and classification information. Sensor nodes only transmit an Acknowledgment Frame when they associate with the mission and match the priority. Table II lists some of the possible missions and the associated devices. Some missions, such as checking for device presence and wireless charging, do not require data retrieval. Thus, the sink returns to the Advertise State when UIWP Ack State ends, shown in Figure 2.

2) UIWP Ack Frame: The UIWP Ack Frame indicates the characteristics of a sensor node, including quantity of data stored, battery voltage and volume, characteristics of antenna, and camera inspection information. The UIWP Ack Frame helps the UAV to determine the corresponding actions. Sensor data is represented by the number of bytes stored which helps the UAV to determine speed and hovering requirements. Battery information may be used to inform wireless charging needs. Antenna information helps the UAV to learn and plan optimal routes, which will be discussed in detail in Section IV-C3.

3) Request Frame: The channel indicated in the data Request Frame informs the sensor node of the data channel it will switch to. The order indicates whether the data starts transmission from the most recent time or the start time, and the bytes indicate how many bytes of data the sink node requests.

4) Data Frame: The Data Frame is divided into data subunits. Each data subunit contains one sensed datum. Data type indicates the types of data, and time stamp records the system time when the data is acquired. Note that a sensor node normally does not have real time clock system. The time stamp records the time elapsed since the node initially boots.

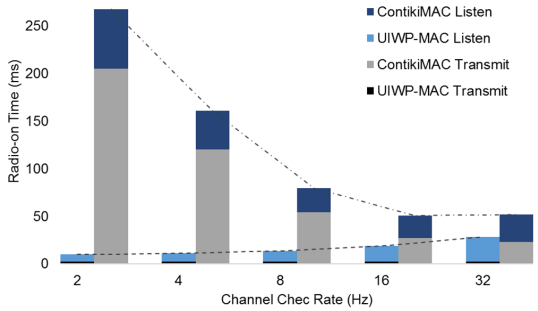

Fig. 5. Per packet radio-on time for different channel check rates at the sensor side. Energy consumption can be estimated by measuring the radio-on time [32].

IV. Evaluation

In this section, the performance of UIWP is evaluated. Initially, energy efficiency and latency of UIWP-MAC are examined. The experimental setups and results are discussed in Section IV-A. Latency, reliability, scalability, and the effectiveness of channel switching of UIWP-APP are evaluated in Section IV-B. In Section IV-C, the impact of the UAV's speed and antenna radiation patterns on data collection performance are evaluated for a realistic outdoor environment.

A. MAC Implementation & Evaluation

UIWP-MAC is evaluated with respect to the energy efficiency and latency. It is implemented using the Contiki OS using the radio duty cycle stack [30], [31]. The Texas Instruments CC2650 LaunchPad is used for the evaluation.

1) Energy Consumption: Micro benchmarks for the energy profile of the static device's radio were determined using software-based energy estimation [32] and physical measurements. Software-based energy estimation results are shown in Figure 5, where energy consumption can be estimated through measuring the radio-on time. The transmission power of the radio was fixed so that the energy consumed can be represented by radio-on time. The average radio-on time for transmitting one packet was measured for 300 packets unicasted to the sink at random intervals, where each packet was 55 bytes in length. Listening time is also measured, which is comprised by periodic channel listening, channel assessment before sending, and acknowledgment wait time.

Figure 5 shows that the average packet transmission time for UIWP-MAC is the same for different channel check rates (CCR). Since the radio of the sink is always on, packets sent by the sensor node are sent only once. This micro benchmarks is shown in Figure 6. The listening time for UIWP-MAC increases with the increment of CCR due to more frequent periodic channel listening. ContikiMAC has gradually reduced radio transmit time as the sink requires less time to detect a packet with increased CCR. The decreasing listening time is due to the reduced acknowledgment waiting time when CCR increases from 2 to 8. Then it increases as the periodic channel listening increases. Figure 5 shows that UIWP-MAC reduces energy consumption of a sensor node significantly, particularly with respect to the transmit energy.

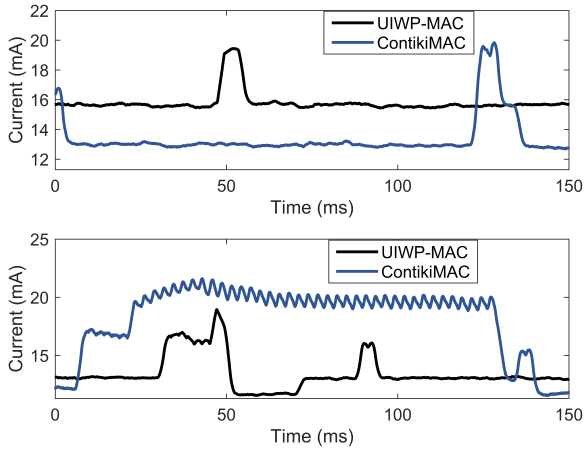

Fig. 6. Per packet energy estimation benchmark. Sink node current profiles are shown in top figure. Sensor node current profiles are shown in bottom figure.

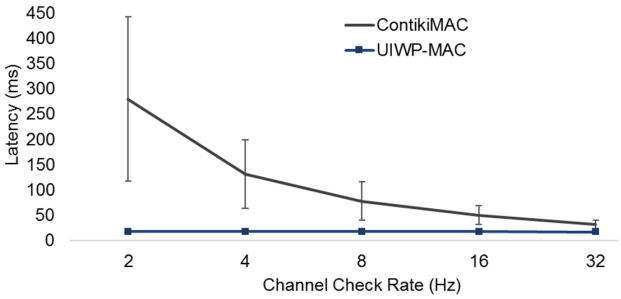

Fig. 7. UIWP-MAC latency for the sensor node. Latency is the time elapsed from the start of packet transmission to physical acknowledgment reception.

For the physical measurements, a 75 Ω resister was connected in series with TI CC2650 LaunchPad, and voltage change was recorded using an oscilloscope. The CCR was set to 8 Hz, which is optimum radio duty cycle without phase lock [25]. The results are plotted for the current change over time. Figure 6 shows the current profile for nodes communicating using UIWP-MAC and ContikiMAC. For UIWP-MAC, the sensor node only send a packet once. In contrast, ContikiMAC sends a packet multiple times until the sink node detects it.

2) Latency: Two types of latencies are measured, for UIWP-MAC and UIWP-APP. The latency evaluated for UIWP-APP is discussed in section IV-B. For UIWP-MAC latency, the time elapsed from the start of sending to acknowledgment reception is measured, which is for the complete cycle of sending a packet. 300 packets of 55 bytes each were unicasted from the static device to a sink at random intervals so as to simulate varied waiting times for packet detection in ContikiMAC. Two consecutive packets were sent at time intervals ranging from 0 to 125 (1 / CCR) milliseconds.

Figure 7 shows the average latencies in milliseconds for UIWP-MAC and ContikiMAC at the static node side. Error bars indicate the standard deviation of the test results. The average latency drops when the CCR increases for ContikiMAC. Compared with UIWP-MAC, the latency is incurred due to waiting for packet detection of the sink.

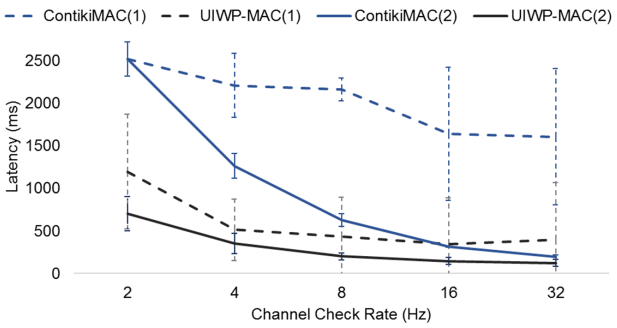

Fig. 8. Average UIWP-APP latency for UIWP-MAC and ContikiMAC. t_{ae} = 2 seconds for (1). t_{ae} are set to 2, 1, 0.5, 0.25, and 0.125 seconds for channel check rate 2, 4, 8, 16, and 32 respectively for (2).

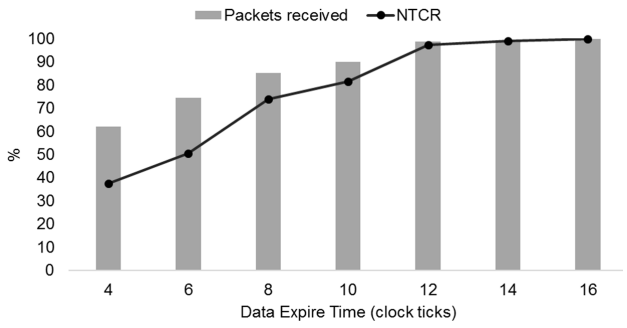

Fig. 9. NTCR; i.e. ratio of the number of times sensor nodes finish uploading periodically collected sensed data to the total attempts, and the percentage of packets received versus total packets sent for increasing t_{de}. 1 clock tick = 7.8125 ms.

B. UIWP-APP Implementation & Evaluation

UIWP-APP is evaluated in terms of latency, reliability, scalability, in addition to examining the effectiveness of the channel switching mechanism.

1) Latency: An experiment was carried out to measure the time elapsed from the first UIWP-APP acknowledgment reception to the first data packet reception; defined as the UIWP-APP latency. This latency is also the time elapsed from the beginning of the UIWP Ack State to the beginning of the Data State, where minimizing this latency accelerates the device discovery process. In comparison with UIWP-MAC, ContikiMAC requires UIWP-APP to set a higher t_{ae} during which to make a decision because of the maximum CCR acknowledgments detection per second. Both ContikiMAC and UIWP-MAC were implemented for UIWP-APP to measure the relative latencies and evaluate the effectiveness of UIWP-MAC.

Three TI CC2650 Sensortags were used as static sensor nodes. The TI CC2650 LaunchPad was used as the sink node. UIWP-APP was tested under two conditions. $N_m = 3$ is set for both cases. For the first case, t_{ae} was fixed for varying CCRs. The fixed t_{ae} is set for $CCR = 2$, where at least 3 channel check periods are required to detect 3 static node acknowledgments. $t_{ae} = 2$ seconds is set to allow one more period for additional checks if collisions happen. For the second case, t_{ae} is set to 4 times the channel check interval with ContikiMAC. UIWP-MAC does not have this maximum detection constraint, hence t_{ae} is set to CCR to allow every neighboring device to detect the Advertise packet. A minimum $t_{ae} = 125$ milliseconds is set for receiving retransmitted packets due to the collision backoff. UIWP-APP latency was tested 100 times for each MAC protocol and each condition. The averaged latencies and standard deviations shown in Figure 8.

UIWP Ack State ends when t_{ae} expires or N_m is reached. For the first condition, a long expiry time t_{ae} causes UIWP-APP with ContikiMAC to have large latency. Since UIWP-MAC collects acknowledgments more efficiently than ContikiMAC, the latency is significantly reduced. For the second condition, latencies are decreased due to shorter t_{ae} for both MACs.

2) Reliability & Data Expiration: Reliability is measured as the node transmission completion rate (NTCR), which is defined as the ratio that the number of times devices finish uploading their periodically sensed data to the total number of attempts. The aim is to understand how t_{de} affects reliability by assessing NTCR.

UIWP-APP was executed with UIWP-MAC for 100 iterations with various t_{de} settings. Three nodes were placed neighboring the sink node. Three types of sensor data were assumed to be acquired every 5 minutes, with the UAV collecting data once daily. Data packets of 96 bytes were used for the experiments, and 72 packets were to be collected from each device each day.

Clock ticks were used to set the data expiry time t_{de}. One second comprises 128 clock ticks for Contiki OS implemented in TI CC2650 chip. A clock tick is thus 7.8125 milliseconds. Shown in Figure 7, the average time for sending a packet from a device is 17.7 milliseconds regardless of the CCR setting. The average time to send a packet is less than 3 clock ticks. However, the experiment showed that when $t_{de} = 3$, NTCR is nearly 0. NTCR recording started when $t_{de} = 4$, as shown in Figure 9. The packets received, as a percentage of total 72 packets for each sensor node, is also given.

It can observed that NTCR and percentage of packets received steadily increase with t_{de}. They reach 100% when $t_{de} = 16$. The extended expiry time increases the chances that the backed-off (i.e. software delayed packets) are received by the sink, thus increasing reliability. However, incrementing t_{de} would delay the sink in terms of realizing if a device is no longer its neighbor.

3) Scalability: As the protocol is designed to have no contention in the Data State, UIWP-APP latency (Section IV-B1) was evaluated for increased neighborhood density. CCR was set to 8 Hz for this experiment. The acknowledgment expiry timer $t_{ae} = 16$ clock ticks and Data Expiry timer $t_{de} = 16$ clock ticks were set. N_m is the number of devices in the sink's neighborhood. 200 iterations were averaged for this test. Figure 10 shows UIWP-APP average latency and standard deviation for varying neighborhood densities, from 1 to 9.

It can be seen that the average latency increases as device density increases to 3. The average latency fluctuates when density continuous to increase. There is no significant

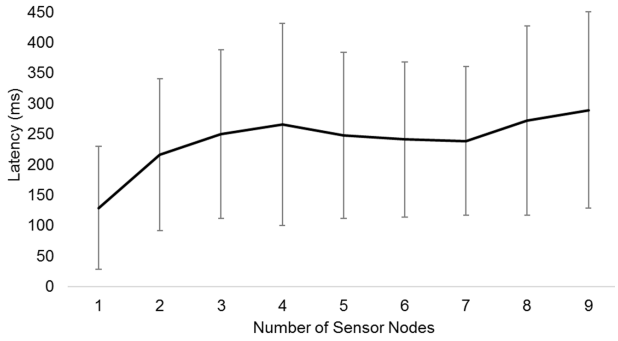

Fig. 10. UIWP-APP latency for increased sensor node densities.

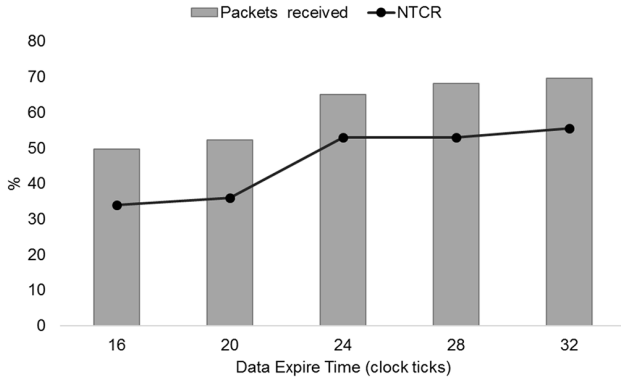

Fig. 11. NTCR and the percentage of packets received when t_{de} increases. NTCR and packets received are the same as described in Figure 9. Channel switching was disabled for this experiment. 1 clock tick = 7.8125 ms.

increment for the latency within the experimented density range, which shows that UIWP works properly in the defined topology.

When there is only one device neighboring the sink, Data Request is immediately sent once the UIWP Acknowledgment packet is received. Therefore, it has the smallest latency. Increasing density increases the time needed to receive N_m acknowledgments. We observe that when the sensor node density is greater than 3, not all acknowledgments can be received each time for $t_{ae} = 16$ clock ticks, and t_{ae} is generally responsible for ending the UIWP Acknowledgment State. By implementing the acknowledgment expiry timer UIWP-APP latency is bounded, which ensures reasonable scalability.

4) Effectiveness of Channel Switching Mechanism: Reliability is used as a metric to evaluate the effectiveness of channel switching by measuring the NTCR and packets received for different data timer expiration settings. From Section IV-B2, it is known that for a 16 clock tick setting with channel switching, all packets transferred can be received. Here, NTCR and packets received without channel switching is evaluated, with all other settings the same as in Section IV-B2.

Experimental results are shown in Figure 11. NTCR and packets received were evaluated with initial setting $t_{de} = 16$. t_{de} was tested up to 32 clock ticks, as this showed the effectiveness of channel switching. It can be observed that

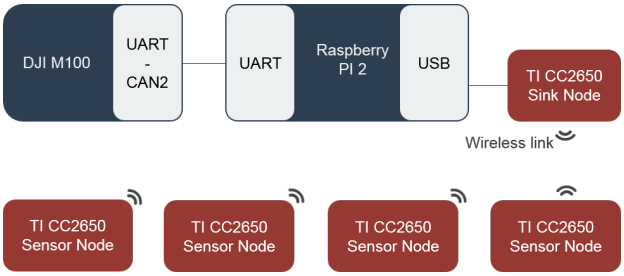

Fig. 12. Schematic representation of the integrated aerial system (TI Sensortag CC2650 - Raspberry PI - DJI M100) used in field experiments.

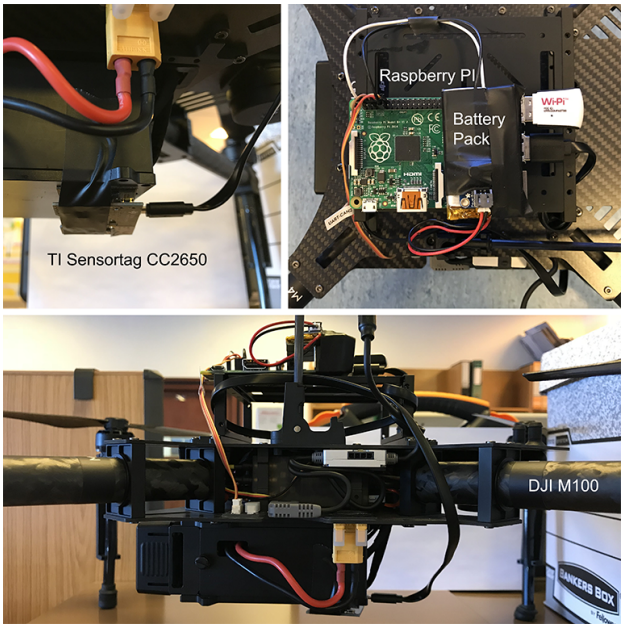

Fig. 13. Top left: TI Sensortag CC2650 attached to the bottom of the UAV. Top right: Raspberry PI with battery pack fixed atop the UAV. Bottom: Side view of the integrated UAV system.

when $t_{de} = 16$, NTCR and packets received are only $1/3$ and $1/2$ compared with using channel switching. Reliability remains insufficient even when doubling t_{de}.

Three sensor nodes were used in this experiment. The NTCR and packets received will become worse as the number of sensor nodes increases due to the increased probability of packet collisions. We expect that the reliability will not become worse if the channel switching mechanism is used, as it is designed to mitigate contention.

C. Field Testing

1) Experiment Settings: Experiments were designed to determine the effects of UAV speed and antenna characteristics on the performance of UIWP. A TI Sensortag CC2650 was used as the sink node connected to a Raspberry PI via a USB cable for data logging, shown in Figure 13. These were mounted on the UAV, with the Raspberry Pi connecting to the DJI M100 via the UART - CAN2 cable for logging purposes. DJI GS Pro App was used for way point-based navigation. A schematic representation of the integrated system is shown Figure 12.

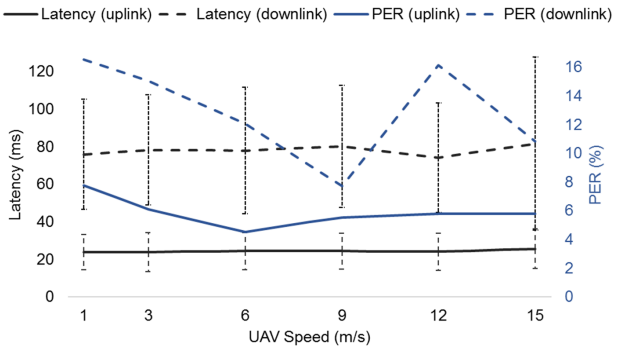

Fig. 14. UIWP-MAC latency and PER for different speeds.

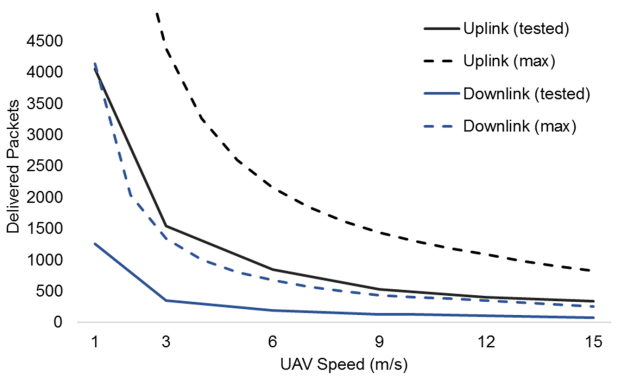

Fig. 15. Delivered packets vs change of UAV speed. The maximum uplink and downlink delivered packets are estimated by assuming the sensor node is always the neighbor of the sink. Delivered packets $N_p = D/t_l$, where D is 314 m and t_l is the averaged UIWP-MAC latency.

2) UAV Speed: The effect of UAV speed on the performance of UIWP-MAC latency, packet error rate (PER), and total packets received when the sensor node was in range of the UAV were evaluated. The uplink and downlink communications were measured individually. For uplink communication, the sensor node continuously sent unicast packets of 55 bytes. The delivered packets and total packets were recorded. The total packets were counted from when the first packet was delivered, i.e. the instant the device enters communication range. The latency was averaged over all delivered packets. A separate test for downlink communication let the sink node continuously send unicast packets, otherwise following the same approach as before.

The experiment was conducted in the Richmond Park, London. The maximum communicable range between two CC2650 transceivers was found to be 100 meters with one on the ground and the other at 10 meters altitude. A straight flying path over the ground node of 314 meters was used, with the ground node placed at the half-way point. This was to ensure that the node could not immediately communicate with the sink. The UAV flew this route repeatedly with different speeds, increasing from 1 m/s.

Figure 14 shows the averaged UIWP-MAC latency and PER for uplink and downlink communications for different speeds, with standard deviation for UIWP-MAC latency. Compared to the relatively slow 1 m/s, increased speed does not significantly influence the UIWP-MAC latency for uplink or downlink. PER jumps abruptly at different speeds. However, it can be observed that packet loss occurs when the UAV is at the periphery of the communication range. There is almost zero packet loss when the UAV approaches directly above the device. Packet loss was measured from when the first packet was delivered. Due to unstable RF conditions at the boundary, PER can change for each measurement.

Figure 15 shows the number of packets delivered for each pass. The inverse relationship is due to the reduced time that the ground node neighbors the sink. Figure 15 shows the maximum uplink and downlink delivered packets, estimated by assuming the node is always a neighbor of the sink. Delivered packets N_p is calculated as $N_p = D/t_l$, where D is 314 m and t_l is the averaged UIWP-MAC latency.

In summary, no significant influence on UIWP-MAC latency was observed for different speeds. However, compared with the static case, UIWP-MAC latency is 5-10 milliseconds longer. Standard deviation is trivial for the static scenario. The packet loss can be up to 10% for uplink communication, owing to unstable radiation at RF boundary.

3) Antenna Characteristics: To understand the effects of antenna radiation patterns on aerial communication with static devices, we firstly evaluated the received signal strength of a device at a fixed distance and for different orientations. The TI Launchpad CC2650 was used for the static device. The Launchpad antenna is shown in Figure 16. The direction of the black arrow in the figure is the 'head' direction, while the opposite is the tail direction.

For received signal strength indicator (RSSI) measurements, the TI Launchpad CC2650 was placed outdoors on the ground. A UAV-mounted TI Sensortag CC2650 acts as the sink node, carried at 10 meters horizontal distance and 1 meter high. The Sensortag and launchpad have the same Inverted F type antenna. The Sensortag's tail direction is always pointed towards the launchpad for these measurements. The head direction of the launchpad was the reference direction; i.e. 0 degrees. The Sensortag continuously sent packets, and the RSSI was recorded at the sink. Due to the changing RF conditions, 50 records were averaged for each point. 15 degree anticlockwise iterations were performed to collect the remaining measurements.

The measurement results in Figure 17 show that the RSSI values fluctuate significantly for different orientations. Received power in some orientations is an order of magnitude lower than for other directions, even when the measurement distance is the same. This indicates that if UAV approaches the sensor node from different angles relative to the 'head', there would be major differences in number of packets transferred.

To better understand the number of packets that can be delivered when the UAV approaches from different orientations, an experiment was conducted; shown on th e right of Figure 16. The UAV was flown four times, from north to south, northwest to southeast, west to east, and southwest to northeast respectively, at a constant speed of 6 m/s. Starting and finishing way points are indicated in the figure. Point C is where the static node was placed. The TI Launchpad CC2650's head

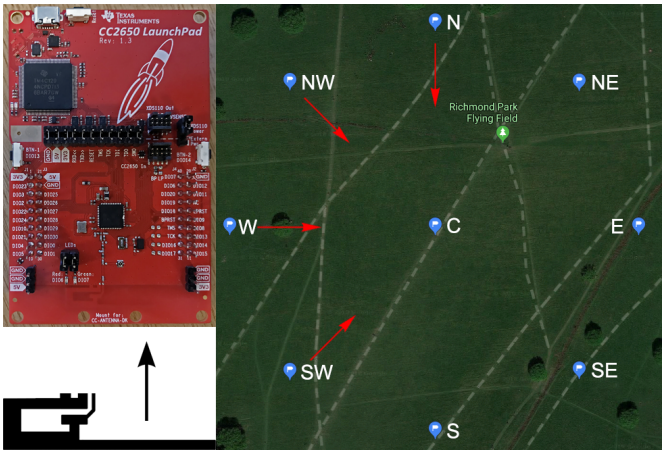

Fig. 16. Left: TI Launchpad CC2650. The direction of the black arrow is defined as the 'head' direction. Right: UAV was flown at a fixed speed of 6 m/s from way point N to S, NW to SE, W to E, and SW to NE at 10, 20, 30, 40 and 50 m. The sensor node was placed at point C. The distance from all way points to point C is ∼159.5 m.

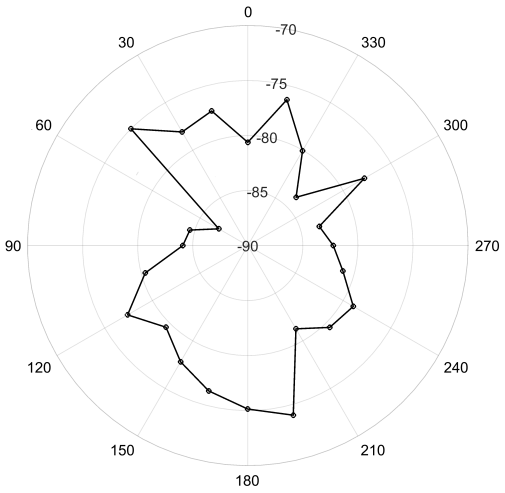

Fig. 17. RSSI value measured at 10 m distance and 1 m altitude.

direction pointed north. For each pass, the tail direction of the sink node always pointed to the static node.

The UAV flew these routes at 10, 20, 30, 40 and 50 meters altitude respectively. The static node constantly sent packets to the UAV. The number of packets received by the sink was recorded for each route and each altitude, the results of which are shown in Figure 18.

Due to the difficulty involved in differentiating the number of packets received from, for example, north to center and center to south, the packets received are divided equally and the plot is symmetric. It can observed that at 30 meters, the number of packets received as the UAV flies from north to south is almost 5 times as much as when flying from west to east. The number of packets received as the UAV flies from north to south achieves its maximum at 30 meters; i.e. above 30 meters, the number of packets received decreases. These results indicate that properly planning both route and altitude

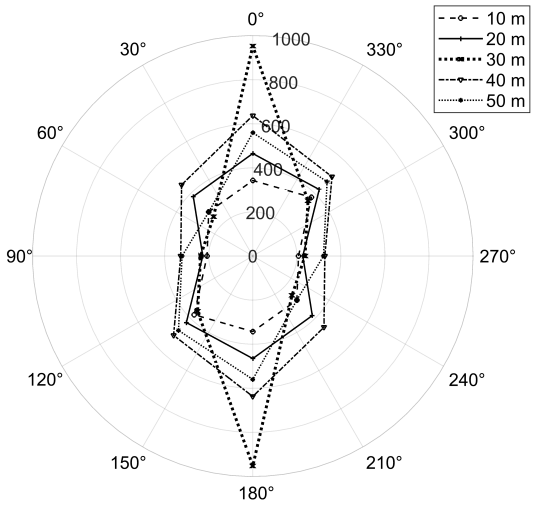

Fig. 18. The number of packets delivered when the UAV was flown N to S, NW to SE, W to E, and SW to NE at different altitudes.

are very important for aerial data collection.

V. Analysis & Discussion

VI. Conclusion

The conclusion goes here.

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 Laksh Bhatia for assisting with the outdoor experiments.

References

[1] Y. Qin, D. Boyle, and E. Yeatman, "A novel protocol for data links between wireless sensors and uav based sink nodes," in *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*. IEEE, 2018, pp. 371–376.

[2] P. D. Mitcheson, D. Boyle, G. Kkelis, D. Yates, J. A. Saenz, S. Aldhaher, and E. Yeatman, "Energy-autonomous sensing systems using drones," in *SENSORS, 2017 IEEE*. IEEE, 2017, pp. 1–3.

[3] J. Valente, D. Sanz, A. Barrientos, J. d. Cerro, Á. Ribeiro, and C. Rossi, "An air-ground wireless sensor network for crop monitoring," *Sensors*, vol. 11, no. 6, pp. 6088–6108, 2011.

[4] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório, and P. A. Vargas, "The use of unmanned aerial vehicles and wireless sensor network in agricultural applications," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*. IEEE, 2012, pp. 5045–5048.

[5] C. A. Trasviña-Moreno, R. Blasco, Á. Marco, R. Casas, and A. Trasviña-Castro, "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring," *Sensors*, vol. 17, no. 3, p. 460, 2017.

[6] H. Chao, M. Baumann, A. Jensen, Y. Chen, Y. Cao, W. Ren, and M. McKee, "Band-reconfigurable multi-uav-based cooperative remote sensing for real-time water management and distributed irrigation control," in *IFAC World Congress, Seoul, Korea*, vol. 17, 2008, pp. 11 744–11 749.

[7] G. Tuna, T. V. Mumcu, K. Gulez, V. C. Gungor, and H. Erturk, "Unmanned aerial vehicle-aided wireless sensor network deployment system for post-disaster monitoring," in *International Conference on Intelligent Computing*. Springer, 2012, pp. 298–305.

[8] M. Erdelj and E. Natalizio, "Uav-assisted disaster management: Applications and open issues," in *Computing, Networking and Communications (ICNC), 2016 International Conference on*. IEEE, 2016, pp. 1–5.

[9] J. Xu, G. Solmaz, R. Rahmatizadeh, D. Turgut, and L. Bölöni, "Animal monitoring with unmanned aerial vehicle-aided wireless sensor networks," in *Local Computer Networks (LCN), 2015 IEEE 40th Conference on*. IEEE, 2015, pp. 125–132.

[10] H. Fu, Z. S. Khodaei, and M. F. Aliabadi, "An event-triggered energy-efficient wireless structural health monitoring system for impact detection in composite airframes," *IEEE Internet of Things Journal*, 2018.

[11] C. Wang, F. Ma, J. Yan, D. De, and S. K. Das, "Efficient aerial data collection with uav in large-scale wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 286080, 2015.

[12] B. Olivieri and M. Endler, "Dadca: An efficient distributed algorithm for aerial datacollection from wireless sensors networks by uavs," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2017, pp. 129–136.

[13] D.-T. Ho, E. I. Grøtli, P. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of wireless sensor network and uav data acquisition," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 1, pp. 159–179, 2015.

[14] I. Jawhar, N. Mohamed, and J. Al-Jaroodi, "Uav-based data communication in wireless sensor networks: Models and strategies," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*. IEEE, 2015, pp. 687–694.

[15] P. Sun, A. Boukerche, and Q. Wu, "Theoretical analysis of the target detection rules for the uav-based wireless sensor networks," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[16] A. Arvanitaki and N. Pappas, "Modeling of a uav-based data collection system," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2017 IEEE 22nd International Workshop on*. IEEE, 2017, pp. 1–6.

[17] S. Rashed and M. Soyturk, "Analyzing the effects of uav mobility patterns on data collection in wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 413, 2017.

[18] Y. Al-Nidawi, H. Yahya, and A. H. Kemp, "Impact of mobility on the iot mac infrastructure: Ieee 802.15. 4e tsch and lldn platform," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 478–483.

[19] H. Li, L. Wang, S. Pang, and M. Towhidnejad, "A cross-layer design for data collecting of the uav-wireless sensor network system," in *Embedded and Ubiquitous Computing (EUC), 2014 12th IEEE International Conference on*. IEEE, 2014, pp. 242–249.

[20] S. Say, H. Inata, J. Liu, and S. Shimamoto, "Priority-based data gathering framework in uav-assisted wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5785–5794, 2016.

[21] A. I. Alshbatat and L. Dong, "Cross layer design for mobile ad-hoc unmanned aerial vehicle communication networks," in *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. IEEE, 2010, pp. 331–336.

[22] Y. Cai, F. R. Yu, J. Li, Y. Zhou, and L. Lamont, "Medium access control for unmanned aerial vehicle (uav) ad-hoc networks with full-duplex radios and multipacket reception capability," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 390–394, 2013.

[23] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noël, "Wireless medium access control under mobility and bursty traffic assumptions in wsns," *Mobile Networks and Applications*, vol. 20, no. 5, pp. 649–660, 2015.

[24] Y. Al-Nidawi and A. H. Kemp, "Mobility aware framework for timeslotted channel hopping ieee 802.15. 4e sensor networks," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7112–7125, 2015.

[25] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[26] S. Duquennoy, A. Elsts, A. Nahas, and G. Oikonomou, "Tsch and 6tisch for contiki: challenges, design and evaluation," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2015)*, 2017.

[27] D. Boyle, R. Kolcun, and E. Yeatman, "Energy-efficient communication in wireless networks," in *ICT-Energy Concepts for Energy Efficiency and Sustainability*. InTech, 2017.

[28] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power coap for contiki," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*. IEEE, 2011, pp. 855–860.

[29] C. Bormann and Z. Shelby, "Blockwise transfers in coap," *draft-ietf-core-block-04 (work in progress)*, 2011.

[30] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.

[31] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 335–349.

[32] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 28–32.