Channel Access Control Protocol for LoRaWAN

Ho Yu Hin
The University of Hong Kong

***Abstract*—LoRaWAN is a low-power wide-area network (LPWAN) technology based on LoRa that uses Aloha-based protocol as its channel access control protocol, which limits its network performance. Several channel access control protocols are implemented in the software simulator extended from LoRaSim and Arduino/Raspberry Pi – based hardware. The results showed that the increase in the number of frequency channel and the use of non-sequential channels improve the network performance in term of packet delivery ratio (PDR). Moreover, the uses of separated channels for the transmission of acknowledgement packets and critical packets reduce the number of retransmissions and improve the PDR of critical packets. By integrating the design in this project to the existing standard, the reliability of the LoRaWAN communication system can be enhanced.**

Keywords—LoRa, LoRaWAN, channel access control, frequency hopping, channel planning

I. INTRODUCTION

LoRa (Long Range) is a state-of-the-art wireless digital modulation scheme with the advantages of long transmission range, low power consumption and high robustness. It operates in the Industrial, Scientific and Medical (ISM) radio bands, 867-869 MHz for Europe, 902-928 MHz for North America and 470-510MHz for China. Using the lowest data rate, it has a transmission range of around 2 km [1]. However, using the lowest data rate yields unacceptable packet delivery ratio (PDR). Hence 1.2 km coverage should be assumed [1]. Moreover, LoRa provides flexibility in varying the physical parameters to fulfill different communication requirements. All the above-mentioned advantages make LoRa a popular communication technology being used in Internet of Things (IoT) applications.

Based on the LoRa technology, LoRaWAN (LoRa Wide Area Network) is one of the LPWAN protocol that is commonly used. However, LoRaWAN only uses Aloha-based protocol and does not provide any channel access control mechanism, which limits its performance in packet transmission [1]. Since the transmission medium is shared in wireless communication, we need an efficient protocol to avoid collision to enhance the PDR. The uses of random frequency hopping (RFH) and carrier sense multiple access with collision avoidance (CSMA/CA) schemes can improve the PDR in the simulation [2]. Another research suggests that it is possible to apply channel planning scheme such as reserving a set of channels for retransmission [3]. Currently, there are few researches focusing on the evaluation of channel access control protocols in both the software simulator and hardware implementation, and they are not adaptive to the LoRaWAN standard. However, it is important to evaluate the performance on both platforms. Software simulator development allows the evaluation of network performance to take place before the implementation of network. It can also be applied in the scalability analysis of the system. Hardware implementation demonstrates the feasibility of implementing different frequency hopping schemes and channel planning schemes in LoRa and how the protocol be compatible with the LoRaWAN standard.

This project aims to evaluate the network performance of the LoRaWAN communication system when different channel access control protocols are applied. Experiments are performed in a simulator developed from LoRaSim and in hardware. The project shows that various channel hopping schemes and channel planning schemes can be implemented compatibly with the LoRaWAN standard and yield improvement in PDR.

The rest of this paper is organized as follows. Section II provides an overview of LoRaWAN. Section III discusses different channel access control protocols. Section IV presents the design of protocols. Section V discusses the functionality added to the simulator and the hardware setup. Section VI provides a performance analysis on the channel access control protocols. Section VII concludes the results in this project and provides various recommendation for future study.

II. OVERVIEW OF LORAWAN

A. Architecture of LoRaWAN

The architecture of LoRaWAN system is shown in Fig. 1. It basically consists of three components, namely sensor/end device, gateway/concentrator and network server.

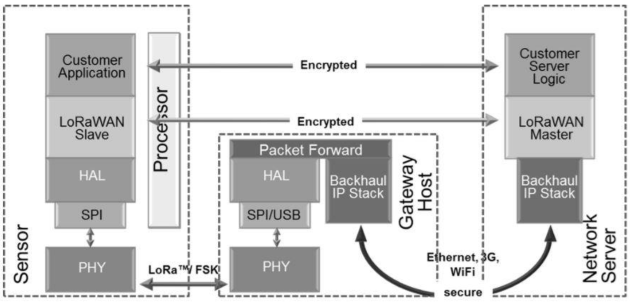

Fig. 1. LoRaWAN architecture

End devices are devices that consist of a microcontroller and a LoRa transceiver. In the microcontroller, there is a microprocessor running customer application and controlling the LoRa transceiver via serial peripheral interface (SPI). In common IoT applications, the customer application collects data from the physical world using sensors and instructs the transceiver to transmit the data as a radio frequency (RF) packet using LoRa modulation or frequency-shift keying (FSK) modulation.

For the application interfacing with the transceiver, there is a hardware abstraction layer (HAL) that is implemented in software for developers to access the transceiver. HAL contains the information of pin mapping and register mapping. This separates the software implementation from the hardware implementation. In other words, changes in hardware selected would not require modification in the application software. This encapsulates the complex hardware details.

The gateway receives the packet from end devices and passes upward to the MAC layer via SPI and HAL. It checks if the packet passes the cyclic redundancy check (CRC) which is an error detecting code. Under common implementation, the gateway only forwards the packet to the network server if the packet passes the CRC. Although it can be set to forward packets that fails to pass the CRC, it is not recommended as most of the network servers are not able to process it.

After the network server has received the packet, it passes the packet to the application as specified in the packet. The network server handle the packets differently depending on the server logic.

The packets are encrypted using AES128 encryption algorithm to ensure security [4]. The keys used in encryption and decryption are determined when an end device joins the network.

There are some classifications of packets defined in LoRaWAN standard [4]. Packets that are sent from the end devices to the network server are called uplink packets. Packets that are sent from the network server to the end devices are called downlink packets. Packets that require an acknowledgement to reply from the receiver are called confirmed packets. Packets that do not requires an acknowledgement to reply from the receiver are called unconfirmed packets. Different types of packets are used in different experiments in this project.

B. Classes of End Devices

Since there are different operation requirements of end devices, LoRaWAN provides three different classes of end devices [4].

Class A devices can only receive data via two receive windows after the transmission as illustrated in Fig. 2. Hence, it is in sleep mode at most of the time, and thus it is the most energy-efficient option. For this reason, it is commonly used in battery powered end devices.

The network server could determine which receive window is used for downlink transmission. Regarding the receive window, it is defined by an offset time, duration and data rate. Receive window 1 (RX1) has an offset of RxDelay1 from the transmit slot. Receive window 2 (RX2) has an offset of RxDelay2 from the transmit slot. The length of RX1 and RX2 must be larger than the time on air of the uplink packet.

The data rate of the first downlink transmission is calculated as a function of the uplink data rate and the receive window offset. The data rate of second window is fixed to be 0.3 kbps.

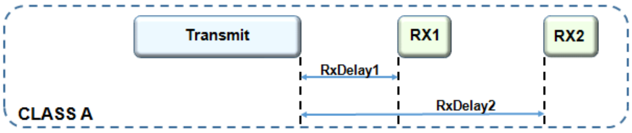

Fig. 2. Class A device timing diagram [4]

Class B is extended from class A and used when the applications need to receive extra downlink data. Class B devices are synchronized by periodic beacons (BCN) sent by the gateway as illustrated in Fig. 3. The gateway can schedule additional receive windows called ping slots for devices. Not only can a class B device receive downlink packets after transmission, it can also receive downlink packets periodically. Since its receiving windows open more frequently than a Class A device, it is less energy efficient.

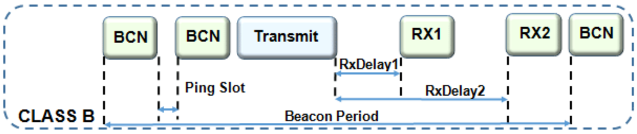

Fig. 3. Class B device timing diagram [4]

Class C is also extended from class A and used in main powered actuators. Class C devices can listen to the channel continuously, except when they are transmitting, as illustrated in Fig. 4. The latency for downlink communication can be minimized. Class C devices are less energy efficient than Class A and Class B devices.

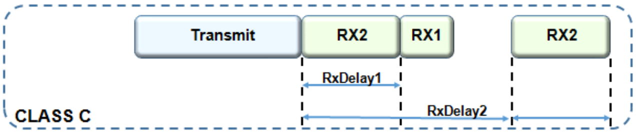

Fig. 4. Class C device timing diagram [4]

Class A must be implemented in all end devices so that they provide the basic functionality of LoRaWAN. Class B and Class C must be compatible with Class A. All end devices must join the network using Class A.

C. Frequency Plan

EU868 frequency band is used in the experiments

Uplink:

1. **868.1** - SF7BW125 to SF12BW125
2. **868.3** - SF7BW125 to SF12BW125 and SF7BW250
3. **868.5** - SF7BW125 to SF12BW125
4. **867.1** - SF7BW125 to SF12BW125
5. **867.3** - SF7BW125 to SF12BW125
6. **867.5** - SF7BW125 to SF12BW125
7. **867.7** - SF7BW125 to SF12BW125
8. **867.9** - SF7BW125 to SF12BW125
9. **868.8** - FSK

Downlink:

- Uplink channels 1-9 (RX1)
- **869.525** - SF9BW125 (RX2 downlink only)

Fig. 5. Frequency Plan [5]

According to the plan listed in Fig. 5, there are eight LoRa uplink channels and one FSK uplink channel. There are eight downlink channels having the same frequency as the uplink channels for receive window 1 and one specific downlink channel for receive window 2. Channel 1-3 are the default channels for uplink defined in the standard. However, they may be disabled for some of the experiments. Channel 8 is disabled in all experiments as FSK is out of the scope of this project.

III. CHANNEL ACCESS CONTROL PROTOCOL

Channel access control protocols can be classified into frequency division multiple access (FDMA), time division multiple access (TDMA), code division multiple access (CDMA), space division multiple access (SDMA) and power division multiple access (PDMA). Those channel access method multiplex the signal in frequency domain, time domain, coding, space domain and power domain respectively. Among these methods, FDMA, TDMA and CDMA are commonly used in IoT communication technology.

Frequency hopping techniques belong to CDMA and are usually called frequency hopping code division multiple access. It can be classified into channel-ignorant or channel-aware. For channel-ignorant frequency hopping, the hopping sequence is determined regardless of channel quality [6]. For channel-aware frequency hopping, the hopping sequence is determined based on channel quality. It can be further classified into reduced-hop-set and probabilistic-channel-usage technique. Reduced-hop-set means that the bad channels are eliminated from the frequency list. Probabilistic-channel-usage means that bad channels would have a lower probability in selection, but not eliminated. Fig. 6 illustrates the classification of frequency hopping techniques.

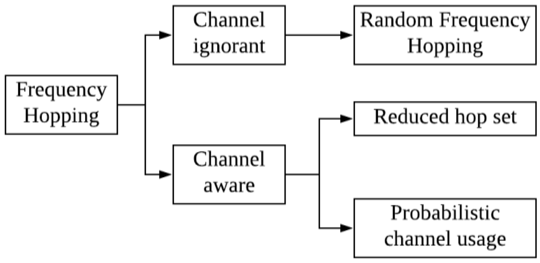

Fig. 6. Classification of Frequency Hopping Techniques [6]

Limited by the scope of the project, channel aware protocols are not investigated in this project.

A. Random Frequency Hopping

A commonly used channel ignorant protocol is random frequency hopping (RFH). RFH is a method of transmitting signal with different frequency channels randomly. A device first randomly or pseudo randomly selects a frequency channel. Then it transmits the packet using that channel. However, the hopping sequence of the channels can be designed to improve the reliability of the system. Also, by planning uplink channels and downlink channels, the PDR of critical packets and acknowledgement packets can be increased. This is the focus of the project.

B. CSMA/CA

Another channel access control protocol is the carrier sense multiple access with collision avoidance (CSMA/CA), which is used in IEEE802.11 RTS/CTS mechanism and IEEE 802.15.4/Wireless PAN standard. A simplest way to describe CSMA/CA is Listen Before Talk (LBT). Before transmission, the sender listens to the channel to check if there is any other device sending packets. If so, it waits a random duration then listens again. It only sends packet if the channel is clear. However, there exists some problems which make it unsuitable to be implemented in LoRa.

The first problem is that determining whether a packet can be successfully received by the gateway is depends on the receiver, not the sender. When an end device listens the channel to check if other end devices are also sending the packet to the gateway, if that two end devices are out of range of each other, they cannot ensure that another device is sending or not, i.e. the hidden terminal problem [7]. Therefore, listening to the channel before sending at the transmitter may not yield a satisfactory improvement, especially for the communication system with large time on air like LoRa.

The second problem is that LoRa chips developed by Semtech do not provide a reliable mechanism to ensure that the channel is not occupied. Since a LoRa transceiver can receive packet below the noise floor, using RSSI to determine whether there is a packet in a channel is not reliable [8]. Therefore, LoRa chips provide a mechanism called channel activity detection (CAD) which is claimed to be able to detect the preamble of the packet [9]. Detection of preamble is obviously insufficient to implement a reliable LBT mechanism as failure of detecting payload of a packet does not guarantee that the channel is clear. Although one research performed an experiment and found that CAD not only can detect preamble, but the whole packet, the research also concluded that the reliability of CAD decreases significantly with increasing transmission range, and this problem exacerbates if the network is congested [9]. Specifically, CAD becomes unreliable even only at a transmission distance of 400m in a dense environment. The failure in detecting the channel activity using CAD leads to difficult implementation of reliable CSMA/CA mechanism on LoRa.

The third problem of CSMA/CA is that it requires the opening of receive windows to detect the activity on the channel. This increases the energy consumption of the devices, which is contradictory to the design principle of a LPWAN, aiming to maximize the life time of battery powered device such that replacement of battery would not occur frequently.

Therefore, CSMA/CA is not implemented in this project. In contrast, the focus of the study is on the frequency hopping schemes and channel planning schemes.

IV. Design of Protocol

A. Frequency Hopping Schemes

There are several ways for devices to hop between different channels.

1. Random Selection
2. Round Robin Scheduling with Sequential Channel
3. Round Robin Scheduling with Non-sequential Channel

For random selection, a random number in range [0, number of channels] is generated by a random number generator before each time of transmission. This gives high level of randomness but requires extra processing. It increases the latency and power consumption. The latency problem is not a major concentration for LoRa-based application. However, extra processing is not necessary.

For round robin scheduling with sequential channel, the channel is selected sequentially in a round robin fashion. For example, 4 channels can be selected, namely, {C1, C2, C3, C4}. The channel is selected sequentially as C1, C2, C3, C4, C1, C2, and so on. This method introduces a forever-collision problem when devices in the network are synchronized and send packets at the same time. If their transmitted packets collide with channel C_j, since the channels in the channel list are sequential, they transmit the next packet using same channel C_{j+1} if $j+1 \leq m$ or C_1 if $j+1 > m$, where m is the largest channel number. That means their next packet definitely collides again.

An example of happening forever-collision problem is that the smart meters application in which end devices report data to the network server simultaneously at a particular moment periodically. Another scenario is the occurrence of sudden incidents. For example, a detector detects that there is a fire or an earthquake. In this case, multiple end devices report the event simultaneously. This leads to high probability of packet collision. In real-life application, there are often other measures to alleviate this problem, such as introducing a random delay before transmission. However, this problem still exists in worst-case scenario. Also, adding random delay is inappropriate for reporting sudden events as the network server needs to be alerted as fast as possible.

For round robin scheduling with non-sequential channel, the channel sequence in the channel list is randomly determined. For example, 4 channels can be selected, namely, {C1, C2, C3, C4}. A channel list is randomly generated, such as {C2, C1, C4, C3}. The channel is selected as C2, C1, C4, C3, C2, C1 and so on. Another end device may have a channel list of {C3, C4, C1, C2} and its channel is selected as C3, C4, C1, C2, C3, C4 and so on. Obviously, this method introduces two benefits. Firstly, extra processing for randomly selecting a channel is not needed. Secondly, this reduces the probability of occurring the forever-collision problem as this problem occurs only when their channel lists are identical. Assume there are N channels. The probability of two end devices having the same channel list is

$$P(\text{same channel list}) = \frac{1}{N!} \tag{1}$$

Therefore, using non-sequential channel greatly reduces the chance of occurring the forever-collision problem.

B. Channel Planning

This section discusses the effect of allocating a specific channel for the transmission of specific types of packets. Two types of packets are discussed in this report, including acknowledgement packets and critical packets.

Since acknowledgement and critical packets are transmitted less frequently comparing to normal packets, it is beneficial to allocate a specific channel for the transmission of acknowledgement or critical packets to reduce the chance of collision by the normal packet.

1) Using separated Channel for Acknowledgement Packet

According to the LoRaWAN standard, there are two types of packet – confirmed and unconfirmed. If the receiver receives a confirmed packet, it must reply an acknowledgement packet to the sender. If the receiver receives an unconfirmed packet, it does not need to transmit any downlink packet. If the sender does not receive an acknowledgement, depending on the application layer implementation, the sender may retransmit the packet until it receives an acknowledgement or gives up after several times of retransmission. The retransmission is detrimental to the network performance as the network is more congested when there are more transmissions.

There are two main reasons of not receiving an acknowledgement packet. First, packet loss occurs in uplink. The network server does not receive the uplink packet from the end device. It can be due to the collision at the gateway, failed CRC or buffer overflow in the gateway. Second, packet loss occurs in downlink. The network server sends an acknowledgement to the end device via the gateway. However, since the receive window has the same frequency as the uplink channel, collision occurs when other end devices or gateways send packet using that frequency. The acknowledgement packet collides with other packets at that end device, making it not able to receive the acknowledgement.

The first reason of packet loss can be mitigated by channel access control protocols or applying different frequency hopping schemes as discussed in section IV A. The second reason can be mitigated by channel planning.

Using a separated channel for transmission of downlink packet can reduce the chance of collision. There are two explanations of the benefits provided from this method. Firstly, the acknowledgement packet at the end device never collide with packets sent from other end devices. Secondly, although the acknowledgement packet may collide with other downlink packet sent from other gateways, the chance is low due to the assumption that the number of gateways is far less than the number of end devices. Recall that a LoRaWAN network has a star of stars topology which means multiple end devices are connected to a single gateway. Since the transmission range of the gateway is large and it can handle thousands of end devices, it is logical to assume that each device is only in the range of few numbers of gateways. Therefore, the probability of collision of downlink packet is low. In the extreme case when an end device is only in the range of a single gateway, it is impossible for the downlink packet collision to occur, since a gateway can only transmit a single packet at a time.

From section II C, it can be observed that there are eight channels for uplink and downlink transmission using LoRa. There is also one channel for downlink for RX2 only. Therefore, the simplest way is to extend the downlink for RX2 to both the RX1 and RX1, since the acknowledgement packet from TTN is sent to the RX1 of the end device. Below shows the modified frequency plan.

From the Fig. 7, it can be observed that the downlink channels of receive window 1 (RX1) are the same as the uplink channels. In other words, if the gateway receives a packet from channel x, the network server replies using channel x. For receive window 2 (RX2), a specific channel with 869.525 MHz, SF9 and BW125 is used.

Since TTN sends downlink packets to the RX1 of end devices, the end devices receive downlink packet from the channel that used to transmit the corresponding uplink packet. To create a specific channel for downlink communication, we can modify the transmitting frequency of the gateway. Regardless of the frequency specified in downlink packets, the transmitter transmit the packets using 869.525 MHz.

Fig. 7. Channel Plan with Separtated Downlink Packets

2) Using separated Channel for Critical Packets

In some applications, there are critical packets of which the delay needs to be minimized. To minimize the chance of collision of critical packets, a separate channel can be allocated for the transmission of critical packets. Critical packets are sent less frequently comparing with normal packets by nature. Therefore, the packet sent in a separate channel is less likely to be collided.

The drawback of using a separate channel for transmission of critical packets is that the network bandwidth is not utilized efficiently. Since critical packets are sent less frequently, the separated channel is unutilized at most of the time. On the other hand, the normal packets share a smaller number of channels, from N to N-1, where N is the maximum number of channels and $N > 2$. The probability of collision increased by a factor of $\frac{N}{N-1}$. The increase is significant if there are few available channels.

It can be observed that there are eight channels for uplink and downlink transmission using LoRa. In this project, channel 8 is reserved for critical packet transmission. In other words, non-critical packets can only be transmitted using channel 1 to 7. The modified channel plan is illustrated in Fig. 8.

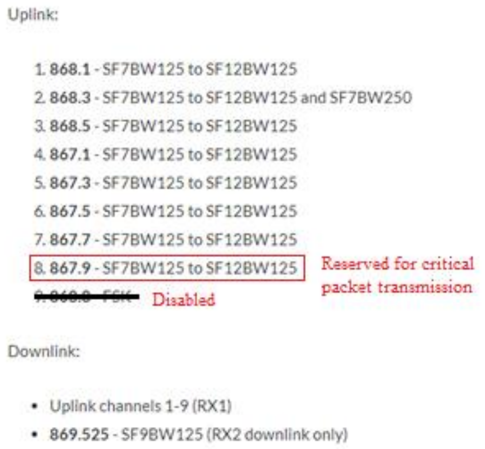

Fig. 8. Channel Plan with Separated Channel for Critical Packets

V. Software and Hardware Implementation

A. LoRa Simulator

Instead of rebuilding a LoRa simulator from scratch, it is built from the LoRaSim, an existing popular LoRa network simulator. The original functionality in the LoRaSim simulator is insufficient for the experiments in this project. Therefore, the following functionalities are implemented.

1. Acknowledgement function
Network server replies an acknowledgement packet if it receives a confirmed packet from an end device.
2. Class A and class B end devices transmission mode as defined in LoRaWAN standard
3. Data buffer in the gateway
The size of data buffer is 1024 bytes as defined in the data sheet of the gateway module. Buffer overflow occurs when a gateway receives more than 1024 bytes of data from the network server.
4. LoRaWAN channel list
5. Shuffling of channel list

B. Hardware setup

There are one gateway and four end devices in this project. Regarding the gateway, Raspberry Pi 2 Model B (RPi2B) is selected as the micro-computer and RAK831 is selected as the LoRaWAN gateway module. Regarding the end devices, Arduino Due is selected as the micro-controller and Dragino LoRa shield is selected as the LoRa transceiver. The reason of choosing these components is justified as follows.

RPi2B provides sufficient computational power while having low power consumption and low-cost. Arduino Due provides sufficient memory for storing the program and provides sufficient computational power. RAK831 and Dragino LoRa shield are popular and low-cost LoRa transceiver modules that exist in the market place.

For the ease of setting up the hardware, RAK2245-RAK831-LoRaGateway-RPi-Raspbian OS developed by the RAK is selected as the operating system of the gateway because the installation procedure is less complicated comparing with other operating systems. The Things Network platform is selected as the network server because it is the most popular platform, Thus, assistance in debugging the system can be received from the community.

For the ease of recording the experimental results, a packet logger is set up in the gateway for confirming the correctness of the results.

VI. Performance Analysis

A. Design of Experiment

Recall that this project aims to evaluate the performance of frequency hopping schemes and channel planning schemes in LoRaWAN. The measure of the network performance is the packet delivery ratio, which defines as the ratio of data packets, excluding the retransmitted packets, passing the CRC received by the receivers to the packets sent by the senders. The number of channels, the type of frequency hopping schemes and channel planning schemes are the independent variables in the experiments. There are six experiments in this project.
1. The Effect of Using Different Number of Channels on the PDR of Uplink Packets
2. The Effect of Round Robin Scheduling with Non-sequential Channel on the PDR of Uplink Packets
3. The Effect of Using Separated Channel for Acknowledgement on the PDR of Acknowledgement Packets
4. The Effect of Using Separated Channel for Acknowledgement with Different Number of Nodes on the PDR of Acknowledgement Packets
5. The Effect of Using Separated Channel for Acknowledgement with Different Number of Gateways on the PDR of Acknowledgement Packets
6. The Effect of Using Separated Channel for Critical Packet on the PDR of Critical Packets

Experiments 1 and 3 are performed in both the software and hardware. Experiments 2, 4, 5 are performed in the software only due to the hardware constraint. Experiment 6 is performed in the hardware only because of the time limit of the project.

Regarding the experimental settings, several data traffic models can be used, including periodic data reporting model and event driven data generation model [1]. The use of data traffic models simulating the real-life application is useful for future deployment.

Periodic data reporting model means that each end device transmits a single packet per day. The experiments in this project that use the periodic reporting model also send a packet once per day. The simulation period is 3 years such that each end device would send around 1000 packets in the simulation. Research states that if all smart meters report the data simultaneously, the packet collision probability is high [1], and this is investigated in experiment 2.

Event driven data generation model means that the data is generated following a Poisson distribution with average λ. The experiments in this project that use the event driven data generation model also send the data following a Poisson distribution. The λ specified in that paper is 5 minutes and 1 hour, which is considered to be impractical for the hardware experiment. Instead, λ is set to be 5 seconds in this project. The experiments end when there are 400 packets received, unless otherwise specified.

These two models are considered to be sufficiently general which can be used to model most of the real-life application. Hence, if the experiments prove that there is improvement on network performance by applying those channel hopping schemes and channel plans, there should also be an improvement in a real-life deployment.

Regarding the experimental setting, unless otherwise specified, {SF7, BW125, CR4/5} is used as this is the default choice in LoRaWAN. The hardware consists of four end devices and one gateway.

Experiment 1 aims to evaluate the effect of using different number of channels on the PDR of uplink packets. It is done on both the software and the hardware. Event driven data generation model is selected. The number of channels is selected from {1, 2, 4, 8} as there are maximum eight uplink channels of LoRa in EU868 band as defined in TTN. Unconfirmed packets are sent by the end devices. The PDR of the uplink packet is measured by counting the number of received and missed packets in TTN console, and the result is verified by the packet logger in the gateway. Also, this experiment also aims to confirm the functionality of the simulator, so that it can be used for the scalability analysis in experiments 4 and 5. The functionality of the simulator is confirmed if the result generated from hardware experiment matches that from the simulator.

Experiment 2 aims to evaluate the effect of round robin scheduling with non-sequential channel on the PDR of uplink packets. It is done on software. The reason of not performing this experiment on hardware is that it is difficult to control the end devices to send packet simultaneously, and hence the worst-case scenario of simultaneous transmission cannot be generated effectively. Periodic data reporting model is selected. Unconfirmed packets are sent by the end devices. The number of channels is selected from {1, 2, 4, 8} as there are maximum eight uplink channels of LoRa in EU868 band as defined in TTN channel list. The PDR of the uplink packet is calculated in the LoRa simulator.

Experiment 3 aims to evaluate the effect of using separated channel for acknowledgement on the PDR of acknowledgement packets. It is done on both the software and the hardware. Event driven data generation model is selected. Confirmed packets are sent by the end devices. There are eight uplink channels as specified in TTN channel list. The downlink channel frequency is set to be 869.525 MHz. For the software, the PDR of the acknowledgement packet is measured in the simulator. For the hardware, the PDR of the acknowledgement packet is measured by counting the number of received and "retry confirmed" packets in TTN channel list, and the result is verified by the packet logger in the gateway. The "retry confirmed" indicates that the end device does not receive an acknowledgement from the network server and thus it retransmits the packet which recognized by the TTN as "retry confirmed" packet. Also, this experiment also aims to confirm the functionality of the simulator, such that it can be used for the scalability analysis in experiment 4 and 5. The functionality of the simulator is confirmed if the result generated from hardware experiment matches that from the simulator.

Experiment 4 aims to evaluate the effect of using separated channel for acknowledgement with different number of nodes on the PDR of acknowledgement packet. This is a scalability analysis of the LoRaWAN system. Due to the limitation on the quantity of hardware, the result can only be simulated in the simulator. Event driven data generation is selected. Confirmed packets are sent by the end devices. The channel list is the same as the experiment 3. The number of end devices is selected from {8, 16, 32, 64, 128, 256, 512, 1024, 2048}. There are four gateways in this simulation. The PDR of the acknowledgement packet is measured in the simulator.

Experiment 5 aims to evaluate the effect of using separated channel for acknowledgement with different number of gateways on the PDR of acknowledgement packet. This is also a scalability analysis of the LoRaWAN system. Due to the limitation on the quantity of hardware, the result can also only be simulated in the simulator. Event driven data generation is selected. Confirmed packets are sent by the end devices. The channel list is the same as the experiment 3. The number of base stations is selected from {8, 16, 32, 64, 128, 256}. The number of end devices is set to be 1024 because it is impractical if the number of end devices is smaller than the number of gateways in the real-life deployment. Therefore, it is set to be four times of the number of gateways, so as to be consistent with the hardware setup in this project. The PDR of the acknowledgement packet is measured in the simulator.

Experiment 6 aims to evaluate the effect of using separated channel for critical packet on the PDR of critical packets. It is done on the hardware only due to time limitation of the project. Event driven data generation model is selected. Confirmed packets are sent by the end devices. There are seven uplink channels, i.e. channel 0 - 6. The channel 7 of frequency 867.9 MHz is reserved for the transmission of critical packets. There are one gateway and four end devices. Device 1, 2, 3 and 4 transmit a critical packet for every 5, 6, 7 and 8 packets respectively. These transmission intervals are selected because we assume that there are four different applications where they have different frequency of transmission of critical packets. The PDR of the critical packet is measured by counting the number of received and missed critical packets in TTN console, and the result is verified by the packet logger.

B. Discussion

1) The Effect of Using Different Number of Channels on the PDR of Uplink Packets

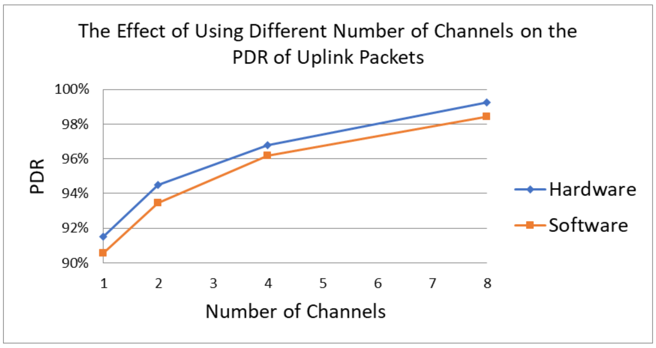

Fig. 9. The Effect of Using Different Number of Channels on the PDR of Uplink Packets

Firstly, it can be observed from Fig. 9 that the increase in number of channels improves the PDR of uplink packets significantly. Specifically, from around 91% for a single channel to around 99% for eight channels.

Also, there exists 1% discrepancy between the result collected from the hardware and the software. This can be due to the difference in the relative distance between the gateway and end devices in the software and in the hardware. End devices are placed randomly and may be as far as 100 meters from the gateway according to the default setting in LoRaSim, whereas end devices are placed at around 0.1 meter to the gateway in the hardware implementation. Therefore, the packets are more easily received by the gateway.

Despite the existence of discrepancy, the result demonstrates that the simulator can correctly simulate the network performance in a real LoRa network.

2) The Effect of Round Robin Scheduling with Non-sequential Channel on the PDR of Uplink packets

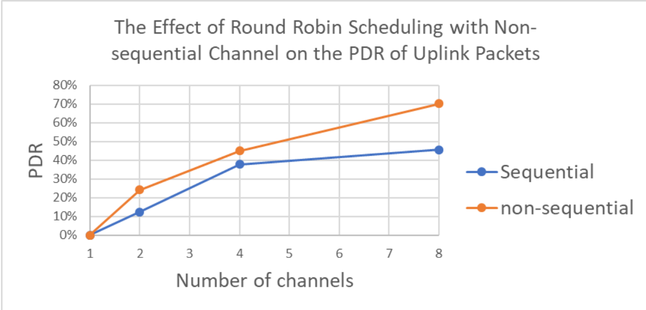

Fig. 10. The Effect of Round Robin Scheduling with Non-sequential Channel on the PDR of Uplink Packets

It can be observed from Fig. 10 that when there is a single channel, the PDR of uplink packets is 0%. This is because a single channel guarantees that the collision happens if the end devices transmit simultaneously. When the number of channels increases, the PDR of uplink packets increases, regardless of whether sequential or non-sequential channel is used. This result is logical as larger number of channels reduces the probability of collision. It also confirms the result in experiment 1.

However, the PDR of uplink packets of using non-sequential channel is higher than that of sequential channel. This is foreseeable as using non-sequential channel reduces the probability of occurring forever-collision problem. Therefore, the use of non-sequential channels improves the PDR of uplink packets in some applications like reporting sudden events.

3) The Effect of Using Separated Channel for Acknowledgement on the PDR of Acknowledgement Packets

Trial / Separated	1	2	3	4	5	Average packet loss	PDR	PDR in simulator
yes	3	8	11	11	5	7.6	98.10%	97.20%
no	23	21	21	20	24	21.8	94.60%	93.50%

Fig. 11. The Effect of Using Separated Channel for Acknowledgement on the PDR of Acknowledgement Packets

It can be observed from Fig. 11 that the use of separated channel for acknowledgement packets yields 3.5% and 3.7% improvement in the PDR of acknowledgement packets for the hardware experiment and simulation respectively. The result is foreseeable since if separated acknowledgement channel is not used, the downlink packet arrives at the end device may collide with other uplink packets sent from other end devices.

Again, the result demonstrates that the simulator can correctly simulate the network performance in a real LoRa network.

4) The Effect of Using Separated Channel for Acknowledgement with Different Number of End Devices on the PDR of Acknowledgement Packets

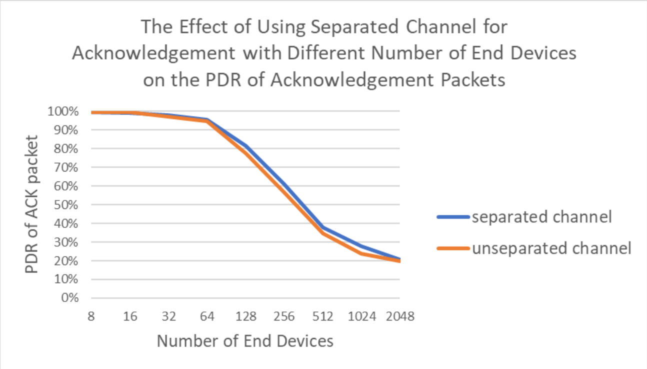

Fig. 12. The Effect of Using Separated Channel for Acknowledgement with Different Number of End Devices

For both schemes, it can be observed that there is a drop in PDR with increasing number of end devices, from 99.5% for eight end devices to around 20% for 2048 end devices. This is because more end devices mean that there are higher chance for the acknowledgement packet to collide with packet sent from the end devices. Also, there is a packet buffer overflow in the gateway when it receives too many packets from the network server.

Separated frequency scheme improves the PDR of acknowledgement packets than unseparated frequency scheme by around 5%. This is because when allocating a specific channel for transmitting acknowledgement, the acknowledgement packets cannot collide with other packets. The probability of collision is lowered and thus PDR increases.

5) The Effect of Using Separated Channel for Acknowledgement with Different Number of Gateways on the PDR of Acknowledgement Packets

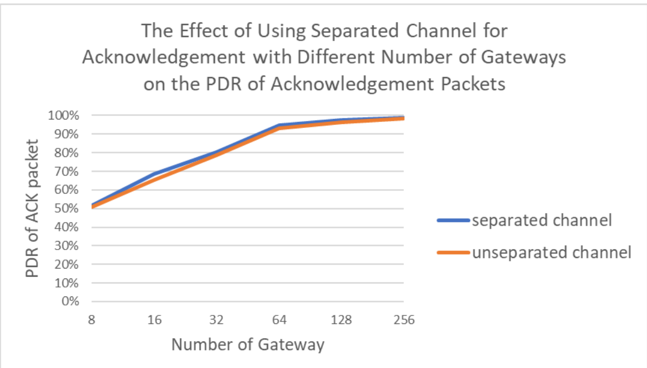

Fig. 13. The Effect of Using Separated Channel for Acknowledgement with Different Number of Gateways on the PDR of Acknowledgement Packets

For both schemes, it can be observed that there is a rise in the PDR of acknowledgement packets with increasing number of gateways. Given that the number of end devices is constant. This is because more gateways mean the packets are distributed to more gateways. The buffer overflow is less likely to occur, and this reduces the chance of ACK packet loss.

Separated frequency scheme improves the PDR of acknowledgement packets than unseparated frequency scheme by around 3%. This is because when allocating a specific channel for transmitting acknowledgement, the acknowledgement packets cannot collide with other packets. The probability of collision is lowered and thus PDR increases.

6) The Effect of Using Separated Channel for Critical Packet on the PDR of Critical Packets and Non-critical Packets

Trial / Separated	1	2	3	4	5	Average critical packet loss	Number of critical packets	PDR of critical packet
yes	3	2	2	4	3	2.8	62	95.50%
no	8	8	7	8	7	7.6	62	87.70%

Fig. 14. The Effect of Using Separated Channel for Critical Packet on the PDR of Critical Packets

The use of separated channel for critical packet yields 7.8% improvement in the PDR of critical packets. The result is foreseeable since when allocating a specific channel for transmitting critical packets, the critical packets cannot collide with other packets. The probability of collision is lowered and thus PDR increases.

Trial / Separated	1	2	3	4	5	Average non-critical packet loss	Number of non-critical packets	PDR of non-critical packet
yes	21	23	24	17	20	21	338	93.80%
no	13	13	11	14	12	12.6	338	96.30%

Fig. 15. The Effect of Using Separated Channel for Critical Packet on the PDR of Non-critical Packets

The use of separated channel for critical packet yields 2.5% decreases in the PDR of non-critical packets. The result is foreseeable since allocating a specific channel for transmitting critical packets would reduce the number of channels that can be used for transmitting non-critical packets. Non-critical packets are more likely to be collided with each other.

However, since the improvement of the PDR for critical packets outweighs the decrease in the PDR for non-critical packets, using separated channel for critical packets is overall beneficial to the network performance.

VII. Conclusion

In this project, the main objective is to evaluate the network performance of the LoRaWAN communication system when different channel access control protocols are applied. The project has fulfilled the two sub-objectives. The first part is the creation of LoRa simulator which is used in the scalability analysis. The second part is to implement different frequency hopping schemes and channel planning schemes on Arduino platform. Both platforms were used in the evaluation of network performance.

Experimental results indicate that the objectives are fulfilled. The simulator developed can successfully simulate the LoRaWAN transmission process in the real world. The results show that the increase in the number of frequency channel and the use of non-sequential channels improve the PDR of uplink packets. Moreover, the uses of separated channels for acknowledgement packet and critical packet transmission reduce the number of retransmissions and improves the PDR of critical packets, respectively.

The main contribution of this project is that it proves that applying frequency hopping schemes and allocating separated channels for acknowledgement packets and critical packets are beneficial to the overall network performance. As the LoRa technology is becoming more popular, more devices will be connected to the LoRaWAN network in the future. With the increasing number of end devices, the probability of collision also increases. Therefore, the channel access control protocols stated in this project is crucial to allow the LoRaWAN communication system to be more reliable and efficient.

However, there are some limitations in this project and exists room for improvement. Due to the limited time, MAC commands were not used for setting up the channels. More channel access control protocols could be tested in LoRaWAN, such as channel-aware frequency hopping or CSMA/CA. Sophisticated channel allocation scheme could be investigated such as distributing the channels to end devices evenly rather than randomly in this project. Also, the extended LoRa simulator based on LoRaSim could further be developed such as implementing class C device or sophisticated traffic model for testing. Due to the limited budget, the scale of the hardware setup was not sufficiently large. If budget allows, experiments could be performed with multiple gateways.

Acknowledgment

I would like to express my gratitude to my supervisor - Prof. Lawrence K. Yeung, who gives me a lot of guidance. In addition, I would like to thank my second examiner – Prof. Ricky Y.K. Kwok, who also provide me feedback in the midterm presentation.

References

[1] M.O. Farooq and D. Pesch, “Evaluation of Multi-Gateway LoRaWAN with Different Data Traffic Models” in *IEEE 43rd Conference on Local Computer Networks*, 2018.

[2] M.O. Farooq and D. Pesch, “A Search into a Suitable Channel Access Control Protocol for LoRa-Based Networks” in *IEEE 43rd Conference on Local Computer Networks*, 2018.

[3] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN," in *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34-40, Sept. 2017.

[4] LoRa Alliance, LoRaWAN 1.1 Specification, 2017.

[5] LoRa Alliance, https://lora-alliance.org/.

[6] E.F.Gorostiza, J. Berzosa, J. Mabe and R. Cortinas, “A Method for Dynamically Selecting the Best Frequency Hopping Technique in Industrial Wireless Sensor Network Applications” *Sensors*, Feb. 2018.

[7] A. Tsertou and D. I. Laurenson, "Revisiting the Hidden Terminal Problem in a CSMA/CA Wireless Network," in *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 817-831, July 2008.

[8] C. Pham, "Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks," *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, 2018, pp. 1-6.

[9] Semtech, LoRa SX1276/77/78/79 data sheet.