# Efficient Image Transmission Using LoRa Technology In Agricultural Monitoring IoT Systems

Tonghao Chen
*Department of Computer Science*
*University of Saskatchewan*
Saskatoon, SK, Canada
tonghao.chen@usask.ca

Derek Eager
*Department of Computer Science*
*University of Saskatchewan*
Saskatoon, SK, Canada
eager@cs.usask.ca

Dwight Makaroff
*Department of Computer Science*
*University of Saskatchewan*
Saskatoon, SK, Canada
makaroff@cs.usask.ca

***Abstract*—Reliable image transmission using LoRa in IoT monitoring systems is considered to be challenging due to insufficient LoRa data rate and payload size. Existing approaches transmit an image in a sequence of packets each of which is individually acknowledged. This approach results in a long image transmission time due to the time spent waiting for the many individual acknowledgements. The acknowledgement traffic also inflates network load. To facilitate LoRa-based image transmission in agricultural monitoring IoT systems, this paper proposes a new reliable delivery protocol, Multi-Packet LoRa (MPLR), for transmission of large messages, such as images, in LoRa networks. The proposed protocol is implemented and evaluated using a LoRa testbed network.**

**In point-to-point experiments with a single sender/receiver pair, MPLR reduced image transmission time by an average of 24% in scenarios with no packet loss, and by averages of 30%, 42%, and 49% in scenarios with 2%, 5%, and 10% loss rate, respectively. When multiple LoRa nodes send images to a single gateway, high channel utilization and an unacceptable collision probability can be experienced with the standard LoRa MAC ALOHA protocol. In experiments with between 5 and 20 nodes, MPLR in conjunction with a channel reservation protocol can successfully send more images and reduce the maximum successful image transmission time between 2 and 7 times, compared to stop-and-wait packet transmission with ALOHA.**

***Index Terms*—LoRa, Wireless Sensor Network, Internet of Things, reliable transport, agricultural IoT systems**

## I. Introduction

The Internet of Things (IoT) has been explored as an effective technology for improving agricultural production management and maintenance [1] [2]. By deploying small battery-powered devices with sensors in agricultural fields, data such as temperature, moisture, and soil nutrient levels can be periodically collected. Each sensed data value can then be sent back to a gateway and eventually uploaded to a server for users to access. Such systems have also greatly lowered the cost of agricultural management and reduced human intervention [3].

The use of wired networks on remote agricultural land is typically not feasible, due to the lack of reliable network and power infrastructure. Personal wireless sensor network (WSN) technologies such as 6LoWPAN [2] and ZigBee [1] [4] [5] have been used in agricultural IoT applications, but these have a signal distance of at most 10 to 20 metres [6] and multi-hop mechanisms are required for larger deployment areas. The range limitations of these personal WSN technologies have prompted much interest in low-power wide-area network (LPWAN) technologies, and in particular LoRa.[1] LoRa-based IoT systems have been used in a variety of agricultural applications, for example smart irrigation [7], soil moisture monitoring [8], rice field management [9], and provision of intelligent agricultural services [10].

Application of IoT systems to the monitoring of crop growth has the potential to increase the efficiency of both crop breeding and production. However, this requires improved capabilities for collection of image data. For example in the *$P^2$IRC* project,[2] researchers are developing crop monitoring and data analysis tools to assist crop breeders. A current focus in this project concerns collection/analysis of images from field test plots. This has motivated our investigation of the possible use of LoRa for image collection.

Image transmission using LoRa is challenging. Due to the limited data rate and payload size, LoRa takes at least 498 seconds to transmit a 1 MB image, using its peak physical layer data rate. This incurs a high energy cost, as well as high network load that could result in a high collision rate using the standard LoRa MAC ALOHA protocol. Furthermore, in some jurisdictions, such as in Europe, there are frequency band duty cycle limits that must be considered [11]. (In other jurisdictions, such as in Canada and the US, other restrictions apply to the frequency band used for LoRa but these do not prevent long-duration or frequent transmissions [11] [12].)

Pham proposed a carrier sense medium access (CSMA) protocol adapted to LoRa networks to avoid packet collisions for image data and transmitted small images through LoRa [13]. Jebril *et al.* demonstrated the concept of point-to-point image transmission using LoRa at a variety of locations and showed that image transmission can be done with delivery times varying between 1 and 14 minutes depending on the spreading factor [14]. To the best of our knowledge, no research has attempted to improve LoRa-based image transmission by replacing the standard LoRa stop-and-wait reliable delivery protocol with a protocol tailored for large messages.

[1] https://www.semtech.com/lora

[2] https://p2irc.usask.ca/

In this study, we propose a novel lightweight and reliable delivery protocol, Multi-Packet LoRa (MPLR). The new protocol batches data packet transmissions and uses bit-vector acknowledgement packets that give the reception status for every data packet within a batch. In comparison to the LoRa stop-and-wait reliable delivery protocol, in which each data packet is individually acknowledged, the new protocol can greatly reduce the acknowledgement traffic and the total time spent waiting for acknowledgements.

MPLR is implemented and evaluated using a LoRa testbed network. In point-to-point experiments with no packet loss, use of MPLR reduced the image transmission time by an average of 24%. In experiments with 2%, 5%, and 10% packet loss rates, average transmission times were reduced by 30%, 42%, and 49%, respectively.

We also propose a data channel reservation protocol for use in scenarios in which multiple LoRa nodes send images to a single gateway. In experiments with from 5-20 nodes, MPLR in conjunction with our channel reservation protocol can successfully send many more images than using stop-and-wait packet transmission and ALOHA as the network becomes congested, and reduces the maximum time to send a single image between 2 and 7 times.

The rest of the paper is organized as follows. Section II introduces background and related work about LoRa. The design and implementation of MPLR and of our channel reservation protocol are described in Section III. Our experimental methodology and evaluation results are presented in Sections IV and V, respectively. Finally, Section VI contains our conclusions and provides directions for future work.

## II. Background and Related Work

The signal range of LoRa is from 10 km to 15 km line-of-sight in rural and 3 km to 5 km line-of-sight in urban areas [15]. Although LoRa has such extensive signal coverage, its power consumption stays low. According to the measurement work done by Cheong *et al.* [16], the power consumption of a LoRa module is 117 $\mathrm{mA}$ in TX mode and is 1.8 µA in sleep mode. Their battery lifetime prediction shows that a battery with a capacity of 2000 mAh ensures a 10-year operational life for a typical LoRa device.

Spreading factor (SF) and bandwidth (BW) are two important factors in LoRa and different combinations of SF and BW lead to a variety of data rates. The physical layer data rate of LoRa is given by the following expression [17]:

$$SF \times \frac{BW}{2^{SF}} \times CR, \tag{1}$$

where SF is the spreading factor, BW is the bandwidth, and CR is the coding rate. Using this equation, the physical layer data rate of LoRa can be calculated for given values of SF, BW, and CR. Table I lists the physical layer data rate of LoRa under all possible combinations of spreading factor and bandwidth when the coding rate is 4/5.

TABLE I: LoRa physical layer data rate (in kbps, CR=4/5)

| | 500 kHz | 250 kHz | 125 kHz |
|---|---|---|---|
| SF7 | 21.88 | 10.94 | 5.47 |
| SF8 | 12.5 | 6.25 | 3.12 |
| SF9 | 7.03 | 3.52 | 1.76 |
| SF10 | 3.91 | 1.95 | 0.98 |
| SF11 | 2.15 | 1.07 | 0.54 |
| SF12 | 1.17 | 0.59 | 0.29 |

Several studies have investigated the applicability and capacity of LoRa to facilitate agricultural management tasks. Zhao *et al.* proposed a smart irrigation system based on LoRa that connects to the Internet at the gateway [7]. Their system provides a convenient method for remote users to send commands and receive status information to/from irrigation nodes. With the aid of LoRa, their system has lower power consumption than a system using GPRS and allows a communication distance between the irrigation node and gateway of up to 8 km.

Payero *et al.* developed and tested a LoRa-based WSN to monitor soil moisture [8]. They constructed several sensor nodes using RFM95 LoRa radios and Decagon EC-5 sensors to collect soil moisture and send data to the coordinating gateway for further aggregation through the Internet. Their system was evaluated in a wheat field and accurately read the value of soil moisture from remote locations.

Ma and Chen used a LoRa-based wireless sensor network to develop a service platform for intelligent agriculture [10]. A multi-sensor component was built and deployed to sense the carbon dioxide concentration, temperature, air humidity, light intensity, soil temperature, soil moisture, wind direction and wind speed from the environment. Collected data was transmitted to a base station through LoRa communication.

Seye *et al.* performed extensive experiments to evaluate the signal coverage and strength of LoRa in Senegal [18] [19]. When using a spreading factor of 12, they observed good signal coverage and strength with a 10 km maximum communication range in urban areas (Dakar, Senegal) and 15 km in rural areas (Namarel, Senegal). Based on these measurements, a proof-of-concept architecture called COWShED was deployed to enable LoRa communication between livestock herders following seasonal migration patterns in Senegal's Ferlo region, which lacks sufficient cellular coverage [19].

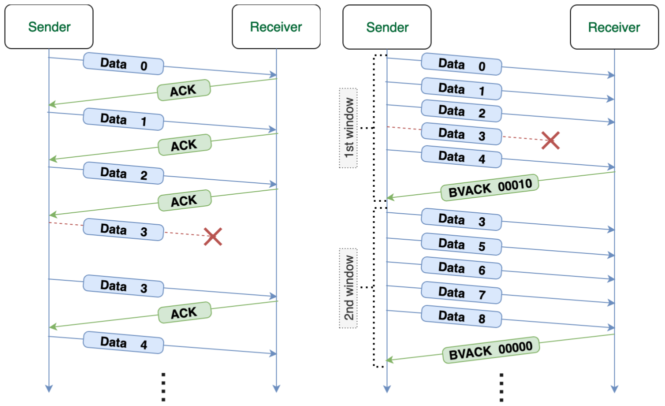
Due to the advantages of LoRa in communication range and power consumption, more possible use cases of LoRa are being explored, including image transmission. Pham divided images of 900 to 1200 bytes into 4 to 5 packets [13]. With the help of their proposed CSMA at the MAC layer, it was feasible to transmit small-sized images when the nodes are hundreds of metres away from each other. Jebril *et al.* used LoRa as the main infrastructure for mangrove monitoring through image transmission [14]. Using a novel image encryption technique, they demonstrated the concept of point-to-point data transmission by transmitting images at various locations.

The objective of this paper is to further enhance the possibility of LoRa-based image transmission in agricultural monitoring IoT systems. We develop a new reliable delivery protocol for transmission of large messages, as well as propose a data channel reservation protocol for scenarios in which multiple LoRa nodes are sending images to a single gateway.

## III. Protocol Design

### A. Multi-Packet LoRa Protocol (MPLR)

Even a highly compressed image is too big to fit in a single LoRa packet, given its maximum payload size of 255 bytes. Therefore, an image must be transmitted using multiple packets. A 9 KB image, for example, needs to be encoded and segmented into at least 48 MTU packets before being transmitted through LoRa. If the stop-and-wait method is used in transmission, as shown in Figure 1a, the sender waits for an acknowledgement per packet to ensure that the data arrives correctly and the packet delivery rate is severely limited. Also, this increases the power consumption of both sender and receiver, and the acknowledgement traffic increases the network load and the required transmission rate of the receiver.



(a) Stop-and-Wait (b) MPLR

Fig. 1: Comparison of packet transmission protocols

Our objective is to reduce the number of acknowledgements that need to be sent and the cumulative time spent waiting for them. For this purpose, we use batched packet transmissions and bit vector acknowledgements. As shown in Figure 1b, according to the size of the transport window, the sender sends a batch of data packets to the receiver consecutively. The correctness of each packet is verified through the forward error correction in the LoRa physical layer and the checksum in the protocol header. The status of each packet delivery is returned to the source via a bit vector acknowledgement packet (BVACK) that contains a bit for every packet in the batch. If the bit in a particular index position is 0, the corresponding packet was received, and 1 if not received. Based on this, the sender can determine which data packets are lost/corrupted. Note that since LoRa employs half-duplex communication, ACKs or NACKs cannot be returned concurrently with data packet transmissions by the sender.

Due to the limited packet payload size in LoRa, our implementation of MPLR uses a lightweight packet format. The packet format is shown in Figure 2.

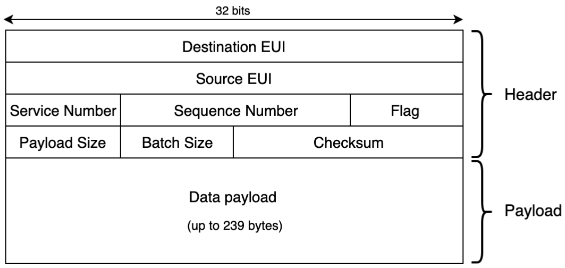


Fig. 2: MPLR message format

The intended packet format consists of the header and data payload. The header has a total of 16 bytes, namely *Destination EUI* (4 bytes), *Source EUI* (4 bytes), *Service number* (1 byte), *Sequence number* (2 bytes), *Flag* (1 byte), *Payload Size* (1 byte), *Batch Size* (1 byte), and *Checksum* (2 bytes). *Destination EUI* and *Source EUI* are the unique identifiers of destination and source device, respectively. The *Service number* identifies the service to which the data packet belongs. The *Sequence number* is used to order the packets. The *Flag* field indicates the packet type. The possible values are *SYN, SYN-ACK, DATA, BVACK, FIN*, and *ACK*. *Payload Size* and *Batch Size* describe the length of the data payload and the size of the current batch, respectively. Finally, *Checksum* is used to check the correctness of the header. The data payload portion still has a space of up to 239 bytes. This header covers the functionalities of both data link and transport layers. The overhead of this 16-byte header is only 6.3%.

Figure 3 illustrates the MPLR protocol for single-hop data transmissions. Similar to many other reliable communication protocols, a connection is first established between the sender and receiver. Then, the sender sends all the packets in the first batch, and waits for a BVACK from the receiver. The sender then transmits the next batch of packets, including, if necessary, retransmissions of packets not correctly received from the previous batch. When all data packets have been sent, the nodes will perform a 4-way handshake through the FIN and ACK packets to complete the current transmission task. The receiver will then return to *LISTEN* mode and wait for the next connection request. The proposed connection and termination method is similar to that used in TCP, but establishment is simplified as the image data is up-link only.

### B. Data Channel Reservation

Although using MPLR can efficiently reduce the image transmission time, it still takes about 7 seconds to transmit a 12 KB image (with SF7/BW500), according to our measurement results in Section V. Compared to sensor data transmission of only a few bytes per message, LoRa-based image transmission has a higher duty cycle and potential for congestion and collisions of request packets when there are multiple nodes using the same gateway.

When sending data packets using the stop-and-wait protocol, collisions due to congestion from data packets from other devices will affect the goodput of the transmission

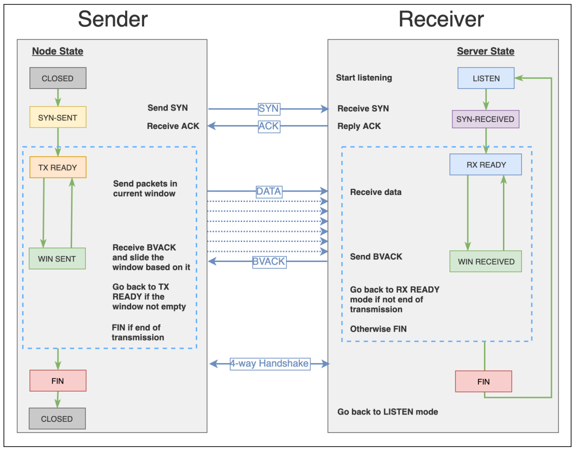


Fig. 3: Connection management: sender/receiver

(rate of correctly received packets). With MPLR, only request packets from other nodes could possibly interfere with the data transmission of a node that has established a connection with the gateway, assuming that the gateway does not accept multiple concurrent connections on the same channel. It is, however, possible to eliminate even these collisions by using a data channel distinct from the control channel.

We therefore design and implement a data channel reservation protocol. Our current gateway only supports listening on a single channel at once, and so our protocol need not support multiple active data channels. However, our approach is easily generalized for scenarios in which the gateway hardware supports concurrent use of multiple channels. Such a generalization is potential future work. As shown in Figure 4, an idle gateway continuously listens for request packets on the control channel. When the gateway receives a request packet, it pseudo-randomly selects a channel to use as the data channel and informs the requesting node of its choice.[3] Then both gateway and node will switch from the control channel to the target data channel and perform the transmission described in the previous section. When the transmission is complete, the gateway will return to the control channel and wait for a new request. Note that any new requests that are made on the control channel while the gateway is listening to the data channel will not be received and will not interfere with the data channel communication. Other devices will not send any packets on any channel other than the control channel until the gateway answers their request.

The data channel selection is not made in an entirely random fashion, since the gateway will avoid choosing channels on which a high loss rate has been experienced. When a particular channel is in use as the data channel, the gateway monitors the packet loss rate on that channel. Such an approach can avoid the use of channels that may be used by other devices in the receiver range or have other reliability issues.

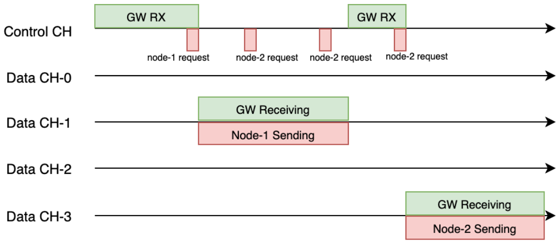


Fig. 4: Data channel reservation timing

Figure 5 shows an example of data channel reservation between the LoRa node and the gateway. The node first sends a *SYN* packet to request connection establishment with the gateway on the control channel. If the request packet is not received by the gateway, the node will sleep for a random time before its next connection attempt. When the gateway receives the request packet, it assigns a data channel and indicates it in the acknowledgement packet. After the gateway replies, it will immediately switch to the assigned data channel and wait for the confirmation from the node. The node will also switch to the assigned data channel after receiving the *SYN-ACK* correctly and send an *ACK* as well.

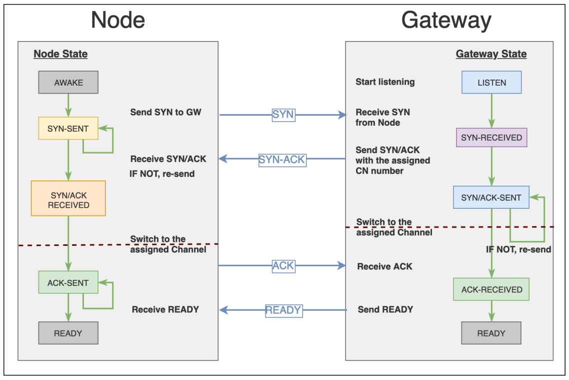


Fig. 5: Data channel reservation: node and gateway

To ensure that the node does not begin transmission before the gateway is ready to begin receiving on the data channel, the gateway will send a *READY* packet to indicate that it is prepared to start receiving data. After the node receives the *READY* packet, it can begin the data transmission. The whole process is a 4-way handshake connection, but using two different channels.

## IV. Experimental Methodology

In this section, the image compression approach is described, and the hardware devices and device settings used in the experiments are introduced. In addition, the experimental plan and topology are discussed.

### A. Image Compression

Images can provide specific information for breeders and producers, such as flowering time and flower density. They can also provide the physical status of the remote IoT system, such as camera/sensor positioning, or physical impediments

[3] In scenarios where frequency hopping is employed, the ”data channel” in our description here would actually correspond to a sequence of channels.

that affect image capture, like weather conditions or wildlife interference. This can be useful for directing system operation. For example, a rainstorm may change the desired data collection parameters for a period of minutes/hours; a human operator or automated adaptation system could change the frequency of image capture or soil moisture reading.

In many such use cases, images can be greatly compressed without impacting their usefulness. In our experiments we use images from a field-deployed camera that generates a 3280x2464 full-resolution image of 5.8 MB. We use Pillow,[4] a Python imaging library, to resize the image to 480x320 pixels, as might be sufficient for a monitoring application, and to apply JPEG compression. An image quality parameter can be passed to the compression algorithm to indicate the degree of compression, which is on a scale from 1 (most compression) to 95 (least compression). This parameter can be tuned depending on application requirements.

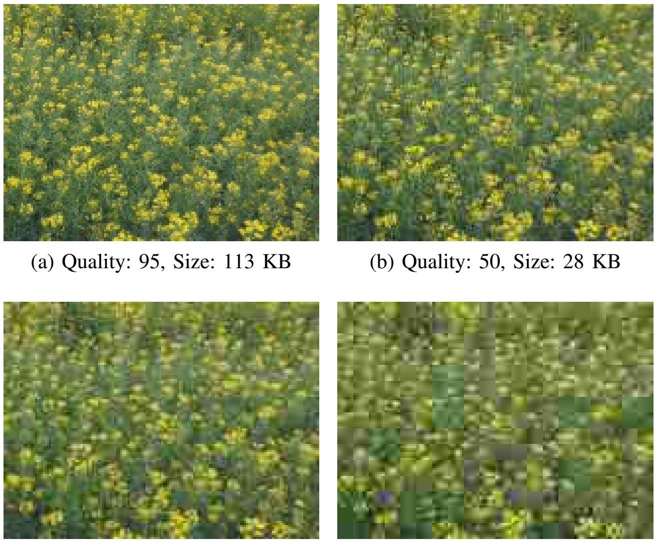
To illustrate the impact of different settings for the image quality parameter, Figure 6 shows results from applying compression to a resized 480x320 pixel image from a canola field. Each subfigure only shows a 160x120 portion from the centre of the image to allow easier visual comparison. The caption for each subfigure gives the corresponding value for the image quality parameter as well as the resulting size of the image in kilobytes. With quality 50 and 25, there is little loss in quality, and the size is reduced to 28 KB and 18 KB, respectively. Significant image distortion is noticeable when the image quality is 10, but this quality may be acceptable in some applications. Thus, by using JPEG compression, a 480x320 resized image compressed to 113 KB (highest quality) can be converted into an image of only 28 KB, 18 KB, or 9 KB, with quality parameter settings of 50, 25, and 10, respectively, with only small to moderate loss of quality.

### *B. Devices*

Our agricultural monitoring IoT system uses the Raspberry Pi (RPI) to sense field data and control the camera. We use the Dragino LoRa hat[5] designed for RPI. This LoRa module is an SX1276 chip-based transceiver[6] and the data communication is done through the serial peripheral interface. As mentioned in the previous section, this transceiver is only capable of listening on one channel at a time. The only difference between gateway and node in our scenario is that the gateway has an Ethernet connection. The field-deployed camera used is the Brinno TLC200 Pro.

### *C. LoRa Configuration*

The LoRa parameter settings used in our experiments are listed in Table II. The output power is set to 15 dBm in all our experiments, and the performance through a range of power levels will be examined in future studies. To implement the stop-and-wait protocol, we add packet headers and enforce acknowledgements in LoRa's physical layer communication.



(a) Quality: 95, Size: 113 KB

(b) Quality: 50, Size: 28 KB

(c) Quality: 25, Size: 18 KB

(d) Quality: 10, Size: 9 KB

Fig. 6: Image quality and size comparison

TABLE II: LoRa configuration

| Name | Value |
|---|---|
| Spreading Factor | 7, 8, 9, 10, 11 |
| Bandwidth | 500, 250, and 125 kHz |
| Coding Rate | 4/5 |
| Implicit Header Mode | false |
| Preamble | 8 |
| Output Power | 15 dBm |

### *D. Deployment and Experimental Parameters*

We conducted two types of experiments to evaluate image transmission in LoRa, namely point-to-point experiments with just a single sender and receiver, and experiments using a star topology. In all experiments, the maximum MPLR batch size was chosen as 40 packets.

*1) Point-to-point Transmission:* We placed two LoRa devices at a fixed distance in the lab, using different combinations of spreading factor and bandwidth to evaluate the feasibility and compare the performance of the image transmission of MPLR and stop-and-wait transmission. The first set of point-to-point experiments was performed with no packet loss. The next set of experiments introduced artificial packet loss of 2% to 10%. In actual IoT system deployments, packet loss may be caused by noise from the environment and transmissions from other devices using the same frequency ranges. We tested all combinations of spreading factor and bandwidth, except for the three lowest data rate combinations, and repeated each measurement 5 times.

*2) Transmission in a Star Network:* We conducted 4 trials of 125 minutes each, for image transmission using MPLR in conjunction with our data channel reservation protocol, and image transmission using stop-and-wait in conjunction with ALOHA. For each trial, a LoRa gateway is placed in the corner of a 200x300 metre test area to receive data and we placed 5, 10, 15 and 20 LoRa nodes randomly in the

[4] https://pillow.readthedocs.io/en/stable/

[5] http://www.dragino.com/products/lora/item/106-lora-gps-hat.html

[6] https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276

test area. All devices used a spreading factor of 8 and a bandwidth of 250 kHz. In each trial, each node generated an image transmission task every 5 minutes. If an image transmission had not completed before a new task at that node was generated, the new task was queued. Image transmission statistics were obtained from the device log. The reception of packets at the gateway was stopped after 125 minutes, even if there were image transmission tasks queued at nodes. Due to the length of the test, we did not perform multiple replications, but consider the transmission of 25 images (125 / 5) per node sufficient to ensure measurement reliability.

### E. *Performance Metrics*

For the point-to-point experiments, the performance metric we consider is the average image transmission time. For the experiments using a star topology, we examine the distribution of the image transmission time, the number of packet collisions, the number of successfully transmitted images, and inter-node fairness.

## V. Evaluation

### A. *Point-to-point Transmission*

We first ran point-to-point experiments with no packet loss and the average transmission times are reported in Table III. To allow easier visual comparison of these results as well as the results in Table IV, table entries with average transmission times between 40 and 60 seconds are shown with light grey shading, and those with times exceeding 60 seconds are shown with dark grey shading.

TABLE III: Transmit times (secs) no packet loss

(a) MPLR: 9 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 4.92 | 9.65 | 19.11 |
| SF8 | 8.54 | 16.90 | 33.62 |
| SF9 | 15.19 | 30.20 | 60.22 |
| SF10 | 27.25 | 54.32 | |
| SF11 | 49.42 | | |

(b) Stop-and-wait: 9 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 7.85 | 13.14 | 23.72 |
| SF8 | 11.99 | 21.28 | 39.64 |
| SF9 | 19.43 | 36.45 | 73.47 |
| SF10 | 33.03 | 63.53 | |
| SF11 | 58.60 | | |

(c) MPLR: 12 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 6.83 | 13.45 | 26.66 |
| SF8 | 11.90 | 22.07 | 46.91 |
| SF9 | 21.18 | 42.13 | 84.03 |
| SF10 | 38.80 | 75.79 | |
| SF11 | 68.95 | | |

(d) Stop-wait: 12 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 11.03 | 18.34 | 33.15 |
| SF8 | 17.08 | 29.64 | 74.11 |
| SF9 | 48.34 | 60.51 | 97.81 |
| SF10 | 62.07 | 88.61 | |
| SF11 | 81.90 | | |

(e) MPLR: 18 KB

| | BW500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 10.02 | 21.98 | 44.06 |
| SF8 | 19.97 | 36.95 | 68.54 |
| SF9 | 30.97 | 64.41 | 128.35 |
| SF10 | 55.54 | 110.69 | |
| SF11 | 100.72 | | |

(f) Stop-and-wait: 18 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 19.33 | 26.77 | 48.29 |
| SF8 | 24.34 | 43.18 | 81.05 |
| SF9 | 59.45 | 73.11 | 152.93 |
| SF10 | 67.06 | 129.16 | |
| SF11 | 130.54 | | |

(g) MPLR: 28 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 15.77 | 31.00 | 64.57 |
| SF8 | 29.01 | 56.35 | 109.13 |
| SF9 | 49.29 | 102.72 | 195.41 |
| SF10 | 88.42 | 181.72 | |
| SF11 | 164.32 | | |

(h) Stop-and-wait: 28 KB

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 25.37 | 42.30 | 90.77 |
| SF8 | 53.57 | 68.44 | 128.37 |
| SF9 | 88.51 | 146.40 | 226.21 |
| SF10 | 106.17 | 215.53 | |
| SF11 | 207.87 | | |

Image transmission times are substantially lower with MPLR than with stop-and-wait. The average reduction in transmission time, over all settings and image sizes, was 24%, with a maximum of 56% and minimum of 8.8%. This reduction in transmission time allows for more spreading factor and bandwidth options when using MPLR, while maintaining an economical operation time and power consumption.

When packet loss is introduced, both protocols degrade, but in different manners. We repeated the point-to-point transmission experiments by adding different artificial packet loss rates to evaluate the change in the transmission time of the 9 KB image for both protocols. Experiments were run with packet loss rates of 2%, 5%, and 10%. Table IV gives the transmission times averaged over 5 measurements for each protocol and parameter setting. Using MPLR, transmission time increases by a percentage only a little greater than the packet loss percentage, since packet losses indicated in each BVACK do not require a timeout for detection, and since packet loss rarely increases the number of batches used to transmit an image (just the size of the last batch). A significant increment is observed when using stop-and-wait, as each packet loss leads to a timeout and retransmission. The reductions in transmission time when using MPLR, averaged over all bandwidth and spreading factor settings, were 30%, 42%, and 49%, for packet loss rates of 2%, 5%, and 10%, respectively.

TABLE IV: Transmit times (secs) packet loss (9 KB image)

(a) MPLR: 2% Loss

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 5.09 | 9.95 | 19.66 |
| SF8 | 8.82 | 17.39 | 34.53 |
| SF9 | 16.09 | 31.04 | 61.84 |
| SF10 | 28.01 | 55.78 | |
| SF11 | 50.78 | | |

(b) Stop-and-wait: 2% loss

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 11.00 | 16.35 | 28.13 |
| SF8 | 16.14 | 24.49 | 44.50 |
| SF9 | 23.85 | 39.98 | 79.73 |
| SF10 | 36.55 | 67.56 | |
| SF11 | 63.54 | | |

(c) MPLR: 5% loss

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 5.30 | 10.36 | 20.47 |
| SF8 | 9.18 | 18.11 | 35.97 |
| SF9 | 16.28 | 32.32 | 64.39 |
| SF10 | 29.18 | 58.10 | |
| SF11 | 52.88 | | |

(d) Stop-wait: 5% loss

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 19.18 | 22.72 | 35.89 |
| SF8 | 23.15 | 33.26 | 56.67 |
| SF9 | 30.92 | 48.25 | 83.75 |
| SF10 | 43.70 | 76.89 | |
| SF11 | 73.71 | | |

(e) MPLR: 10% loss

| | BW500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 5.51 | 10.77 | 21.28 |
| SF8 | 9.54 | 18.83 | 37.40 |
| SF9 | 16.93 | 33.61 | 66.96 |
| SF10 | 30.33 | 60.40 | |
| SF11 | 54.98 | | |

(f) Stop-and-wait: 10% loss

| | BW 500 | BW 250 | BW 125 |
|---|---|---|---|
| SF7 | 25.45 | 25.45 | 45.25 |
| SF8 | 29.23 | 39.97 | 60.33 |
| SF9 | 35.74 | 56.52 | 93.68 |
| SF10 | 51.90 | 86.12 | |
| SF11 | 81.78 | | |

### B. *Transmission in Star Network*

We report the transmission time for each node sending a single image to the gateway in a network with different numbers of LoRa nodes. Figure 7 shows the distribution of the transmission times for each 9 KB image (using a spreading factor of 8 and bandwidth of 250 kHz) with 5, 10, 15, and 20 nodes using MPLR in conjunction with our data channel reservation protocol, and stop-and-wait in conjunction with ALOHA. The box plots only include successful transmissions, as Section IV-D indicates the effect of backlog on sending.

When using MPLR+channel reservation, the average image transmission time (time from when an image transmission task is generated at a node until the transmission is successfully completed) was 55.6, 104.5, 177.8, and 304.8 seconds, with

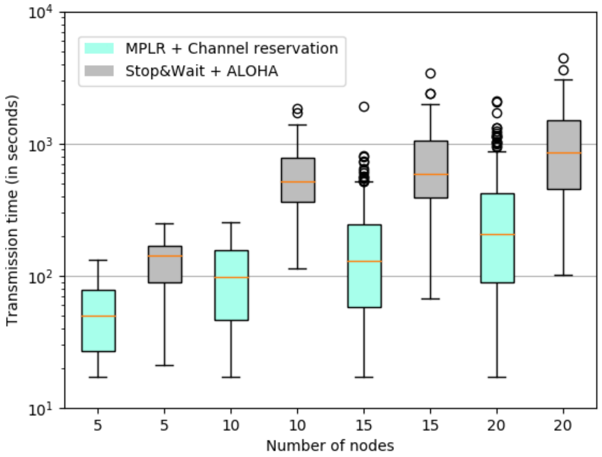


Fig. 7: Image transmission time distribution

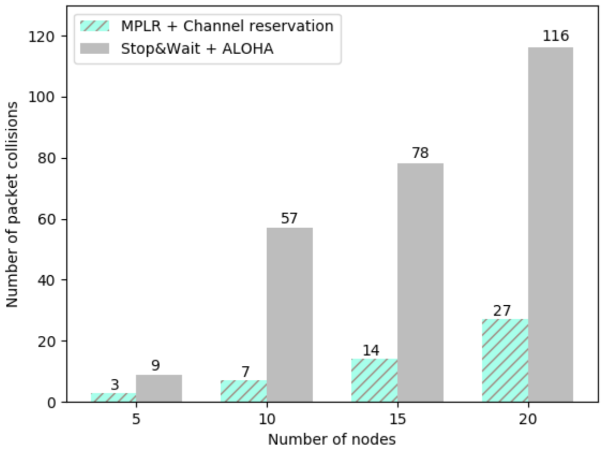


Fig. 8: Packet collisions

5, 10, 15, and 20 nodes, respectively. Note that the image from the first node to successfully reserve the data channel is always delivered after about 17 seconds, with transmissions from other competing nodes then being delayed. When using stop-and-wait+ALOHA, the average transmission time increases greatly when the number of nodes in the network grows, due to collisions among data packets. Even if only 5 nodes are in the network, using stop-and-wait+ALOHA requires an average of 132 seconds to send an image, about 2 times longer than MPLR+channel reservation.

In the case of 5 nodes, the transmission time for the last image to be successfully received at the gateway was about 133 seconds when using MPLR+channel reservation, but was about 250 seconds with stop-and-wait+ALOHA. Scaling up to the 20 node scenario, the last image to be successfully received when using MPLR+channel reservation, within the 125 minute experiment duration, had a transmission time of about 35 minutes (since in this scenario there is substantial queueing of the transmission tasks), while the last image to be successfully received when using stop-and-wait+ALOHA had a transmission time of about 74 minutes. With 10 nodes, stop-and-wait+ALOHA was over 7 times slower than MPLR+channel reservation (1850 vs. 250 seconds).

Figure 8 shows a comparison of the average number of packet collisions during a single image transmission. As the number of nodes in the network increases, the number of packet collisions increases with both methods. However, MPLR+channel reservation produced fewer packet collisions under the same number of LoRa nodes.

Figure 9 depicts the number of images received by the LoRa gateway for each network size. Since an image transmission task is generated at each node every 5 minutes in these experiments, the gateway should receive in total 125, 250, 375, and 500 images, respectively. When there are only 5 nodes in the network, the gateway always receives 125 images. When the number of LoRa nodes in the network is increased to 10, only 114 images are successfully received using stop-and-wait+ALOHA, prior to the experiment termination after 125 minutes, due to congestion and corresponding packet loss; the gateway receives all images in the network using MPLR+channel reservation.

Similarly, when the number of nodes in the network is increased to 15 and 20, when using MPLR+channel reservation the gateway receives 334 and 353 images, respectively, while with stop-and-wait+ALOHA the number of delivered images remains at approximately 120. With MPLR+channel reservation, the network reaches a plateau more gradually than with stop-and-wait+ALOHA. Although this transmission frequency is much higher than the anticipated frequency of demand in actual projects, it allows us to determine the capacity when using MPLR+channel reservation.

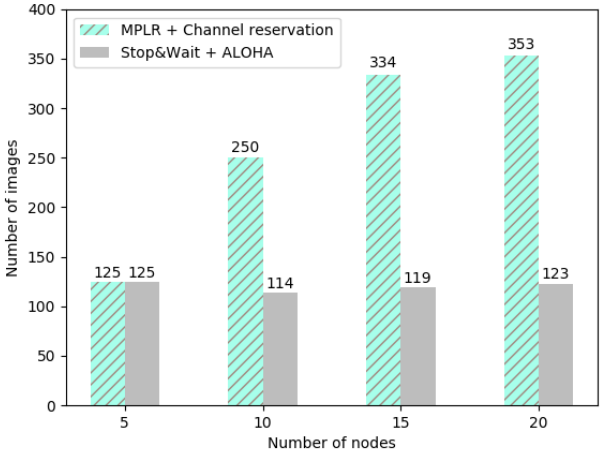


Fig. 9: Successful image transmissions

The image delivery fairness performance difference between MPLR+channel reservation and stop-and-wait+ALOHA is presented in Figure 10. As the number of nodes increases beyond 5, they are differentially affected by the added traffic and potential congestion. With stop-and-wait+ALOHA and 20 nodes, one node is limited to one successful image transmission. This is 25% of the median for this configuration, indicating there are other factors that influence which nodes suffer most from collision.

Under MPLR+channel reservation, the distribution of the number of images successfully delivered for each node is considerably more condensed. 50% of the nodes are no more than 2 images off from the median with 20 nodes and the range is even tighter for 15 nodes. The most extreme outlier for the 20-node scenario can still deliver 13 images (72% of the median).

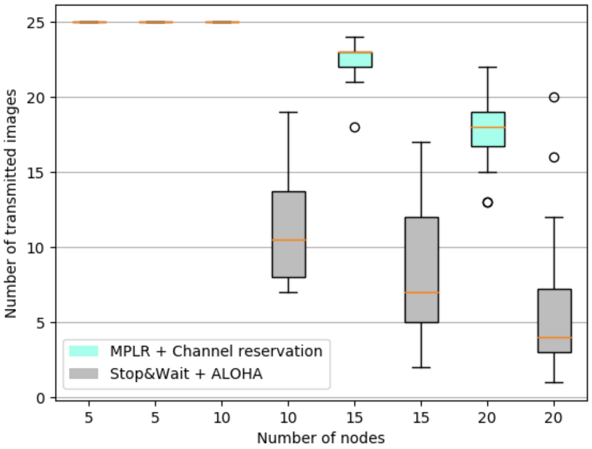


Fig. 10: Inter-node fairness

## VI. Conclusion and Future Work

In this paper, we proposed a lightweight reliable delivery protocol called MPLR for image transmission in LoRa to enable remote image monitoring in our agricultural IoT system. MPLR batches data packet transmissions and uses bit-vector acknowledgements so as to greatly reduce the number of required acknowledgement packets and the time spent waiting for them. To avoid packet collision caused by congestion, data channel reservation is used so that the request packet and data packet can be transmitted on different channels to ensure the sustained successful data transmission rate.

The results show that using MPLR for image transmission in point-to-point communication can reduce the time by an average of 24%. When packet loss was introduced, the average transmission time is at least 30% improved over stop-and-wait and as much as 49% improvement is experienced (for 10% packet loss). In the star network experiments, a shorter average transmission time was experienced with MPLR+channel reservation, and there is no increase in delivery time of the first image with increased network density. Compared with stop-and-wait+ALOHA method, MPLR+channel reservation achieves a higher fairness among the participating nodes.

For future work, we plan to test MPLR+channel reservation in a broader area to further evaluate its performance. In deployments over large areas, error rates can be expected to be significant. Experiments in such settings can provide additional insight into the impact of LoRa parameter values and the value for the maximum MPLR batch size. Also, another ongoing study is to apply the multi-packet LoRa protocol to a multi-hop LoRa network. Although LoRa has a long transmission distance, its signal is easily blocked by obstacles and uneven topography that is likely in a general deployment environment. Multi-hop transmission can solve the issue of signal blocking and further extend the range of the LoRa network.

## References

[1] J. Li and Y. Zhang, “Design and accomplishment of the real-time tracking system of agricultural products logistics process,” in *2010 International Conference on E-Product E-Service and E-Entertainment*, Henan, China, Nov 2010, pp. 1–4.

[2] Z. Suryady, M.H.M. Shaharil, K.A. Bakar, R. Khoshdelniat, G.R. Sinniah, and U. Sarwar, “Performance evaluation of 6LoWPAN-based precision agriculture,” in *ICOIN 2011*, Barcelona, Spain, Jan. 2011, pp. 171–176.

[3] Y. Gu and T. Jing, “The IoT research in supply chain management of fresh agricultural products,” in *AIMSEC 2011*, Dengleng, China, Aug 2011, pp. 7382–7385.

[4] G. Sahitya, N. Balaji, C. D. Naidu, and S. Abinaya, “Designing a wireless sensor network for precision agriculture using ZigBee,” in *2017 IEEE 7th International Advance Computing Conference (IACC)*, Hyderabad, India, Jan. 2017, pp. 287–291.

[5] K. V. de Oliveira, H.M. Esgalha Castelli, S.J. Montebeller, and T.G. Prado Avancini, “Wireless sensor network for smart agriculture using ZigBee protocol,” in *2017 IEEE First Summer School on Smart Cities (S3C)*, Natal, Brazil, Aug. 2017, pp. 61–66.

[6] I. Howitt and J. A. Gutierrez, “IEEE 802.15.4 low rate - wireless personal area network coexistence issues,” in *IEEE Wireless Communications and Networking*, New Orleans, LA, Mar. 2003, pp. 1481–1486.

[7] W. Zhao, S. Lin, J. Han, R. Xu, and L. Hou, “Design and implementation of smart irrigation system based on LoRa,” in *2017 IEEE Globecom Workshops (GC Wkshps)*, Singapore, Singapore, Dec. 2017, pp. 1–6.

[8] J. O. Payero, A. M. Nafchi, R. Davis, and A. Khalilian, “An Arduino-based wireless sensor network for soil moisture monitoring using Decagon EC-5 sensors,” *Open Journal of Soil Science*, vol. 07, no. 10, pp. 288–300, 2017.

[9] T. Hirata, K. Terada, M. Toyota, Y. Takada, K. Matsumoto, and M. S. Tanaka, “Proposal of a power saving network for rice fields using LoRa,” in *GCCE 2017*, Nagoya, Japan, Oct. 2017, pp. 1–4.

[10] Y. Ma and J. Chen, “Toward intelligent agriculture service platform with LoRa-based wireless sensor network,” in *IEEE ICASI*, Chiba, Japan, Apr. 2018, pp. 204–207.

[11] D. Castells-Rufas, A. Galin-Pons, and J. Carrabina, “The regulation of unlicensed sub-GHz bands: Are stronger restrictions required for LPWAN-based IoT success?,” *CoRR*, vol. abs/1812.00031, 2018.

[12] Innovation, Science and Economic Development Canada, “Digital Transmission Systems (DTSs), Frequency Hopping Systems (FHSs) and Licence-Exempt Local Area Network (LE-LAN) Devices,” 2017.

[13] C. Pham, “Robust CSMA for long-range LoRa transmissions with image sensing devices,” in *2018 Wireless Days (WD)*, Dubai, United Arab Emirates, Apr. 2018, pp. 116–122.

[14] A.H. Jebril, A. Sali, A. Ismail, and M.F.A. Rasid, “Overcoming limitations of LoRa physical layer in image transmission,” *Sensors (Basel)*, vol. 18, no. 10, 3257, pp. 1–22, Sept. 2018.

[15] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[16] P. S. Cheong, J. Bergs, C. Hawinkel, and J. Famaey, “Comparison of LoRaWAN classes and their power consumption,” in *2017 IEEE Symposium on Communications and Vehicular Technology (SCVT)*, Leuven, Belgium, Nov. 2017, pp. 1–6.

[17] A. Augustin, J. Yi, T. Clausen, and W.M. Townsley, “A study of LoRa: Long range & low power networks for the Internet of Things,” *Sensors (Basel)*, vol. 16, no. 9, 1466, pp. 1–22, Sept. 2016.

[18] M. R. Seye, B. Gueye, and M. Diallo, “An evaluation of LoRa coverage in Dakar Peninsula,” in *IEEE IEMCON*, Vancouver, Canada, Oct. 2017, pp. 478–482.

[19] M. R. Seye, M. Diallo, B. Gueye, and C. Cambier, “COWShED: Communication within white spots for breeders,” in *IEEE ICIN*, Paris, France, Feb. 2019, pp. 1–4.