International Journal of Control and Automation
Vol. 11, No. 4 (2018), pp.143-154
http//dx.doi.org/10.14257/ijca.2018.11.4.13

Reliable Multi-Hop Linear Network Based on LoRa

Cong-Tan Duong[1] and Myung-Kyun Kim[2*]

[1]44610 Dept. of IT Convergence, Univ. of Ulsan, Daehak-Ro 93, Nam-Gu, Ulsan, Korea
[2]Dept. of IT Convergence, Univ. of Ulsan
[*]mkkim@ulsan.ac.kr

Abstract

LoRa technology enables long range, low power consumption and secure data transmission features. By multi-hop communication, we argue that LoRa is potentially very useful for the monitoring system for wide area with long distance. In this paper, we propose a protocol with reliable multi-hop communication for linear network including network constructing period and data operating period using LoRa. Also, we analyze and show the tradeoffs among synchronization time interval, clock drift rate and configurations in LoRa. Our experimentation using five nodes can form a linear network and delivery reliability achieves above 95% in case of adapting one-time retransmission.

Keywords: *LoRa, Linear Network, IoT, Reliable Multi-hop Communication*

1. Introduction

New Low-Power Wide-Area Network (LPWAN) technologies [10] such as LoRa[9], Sigfox[8], RPMA[7] and Weightless are emerging technologies which allow power efficient wireless communication over very long and wide distances. These technologies target applications where thousands of devices are used in a large geographic area to collect sensor readings. These systems are encountered in a setup where sensor devices send data in one hop to powerful receiver which then forwards data over a fixed wired infrastructure to a data collection point.

In addition to the improved communication range, the LPWAN technologies have unique features stemming from the used modulation schemes. Therefore, to simply adapt these LPWAN technologies, the existing Medium Access Control (MAC) protocols and routing mechanisms are not efficient. Particularly for the monitoring application in which object is required to monitor layout as line style and require increasing coverage long distance.

In this paper, we use LoRa as technology for building an IoT system. We propose a protocol with multi-hop transmission for linear network including network construction period and data operating period. It is suitable for monitoring application in which need coverage long distance and every monitored point is almost placed as line style such as gas line system, high voltage line system, *etc*. Also, we describe a mechanism for wake-up time calculation used in protocol and show the constraints among synchronization time interval, clock drift rate and configurations in LoRa. To evaluate performance of protocol, we perform an experimentation using 5 LoRa nodes (1 sink node and 4 sensor nodes) equipped with a Semtech SX1272 transceiver.

The remainder of this paper is organized as follows. Some backgrounds about LoRa are given in Section 2. The design of protocol is presented in detail in Section 3. Section 4

Received (January 4, 2018), Review Result (March 19, 2018), Accepted (March 21, 2018)
[*] Corresponding Author

ISSN: 2005-4297 IJCA

evaluates the performances of protocol base on result of experimentation. Finally, we conclude the paper.

2. Background

2.1. LoRa Specification

LoRa [2][3] is a proprietary spread spectrum modulation technique by Semtech. It is a derivative of Chirp Spread Spectrum(CSS) with integrated Forward Error Correction (FEC). A LoRa transceiver can decode transmissions 19.5 dB below the noise floor, thus, enabling very long communication distances. LoRa key properties consist of long range, high robustness, multipath resistance, doppler resistance and low power. Today, LoRa transceiver can operate between 137 MHz to 1020 MHz, and thus can also operate in licensed bands. However, they are often deployed in ISM bands (EU: 868MHz and 433MHz, USA: 915MHz and 433MHz). But the use of frequency is depended on the region (EU:868 MHz-870 MHz, US: 902 MHz-928 MHz, Korea: 917MHz – 923 MHz).

A typical LoRa radio provides five configuration parameters: carrier frequency, bandwidth, transmission power, spreading factor and coding rate [3].

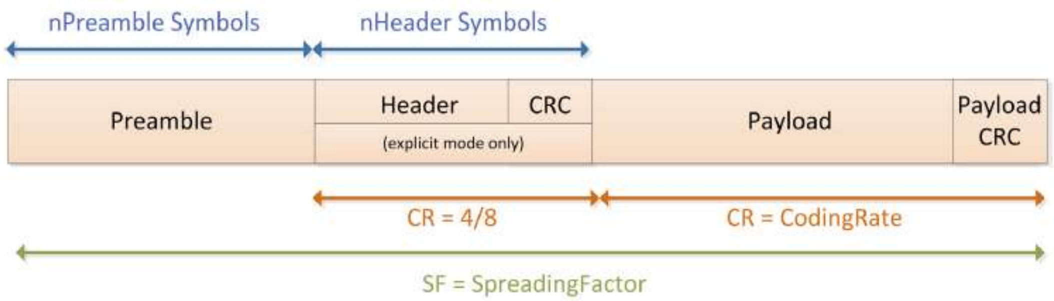


Figure 1. LoRa Packet Structure

The LoRa packet structure is shown in Figure 1. Starting of packet is preamble, it can be programable 6 to 65535 symbols, to which the radio appends 4.25 symbols. In fact, the receiver need at least 4 symbol time to acquire lock on a preamble, so the range of preamble as shown is reasonable. Thereafter is an optional header. There are two type of header mode: explicit mode and implicit mode. With explicit mode, header provides information on the payload including payload length, forward error correction code rate and the presence of an optional 16-bits CRC for the payload. In term of implicit mode, the payload length, coding rate and CRC presence are fixed or known in advance. They are must be manually configured on both sides of radio link. At the end of the payload an optional 16-bit CRC may be included.

The airtime of a LoRa transmission depends, besides the payload size and preamble length, on the combination of SF, BW, and CR. The duration of a transmission can be calculated based on formula which is describes in the datasheet [5] of Semtech LoRa chip in detail. It has to be noted that depending on the selected communication settings a data packet can have significant variations in airtime.

2.2. IBM LoRaWAN C-library (LMIC)

In order to provide a portable implementation of LoRaWan MAC layer protocol, IBM developed the LoRaWAN in C (LMiC) library [11][12]. The architecture of LMiC is divided into four layers: Application layer (APP), MAC state engine

(MAC), Run-time environment (RTE), and Hardware Abstraction Layer (HAL), as shown in Figure 2.

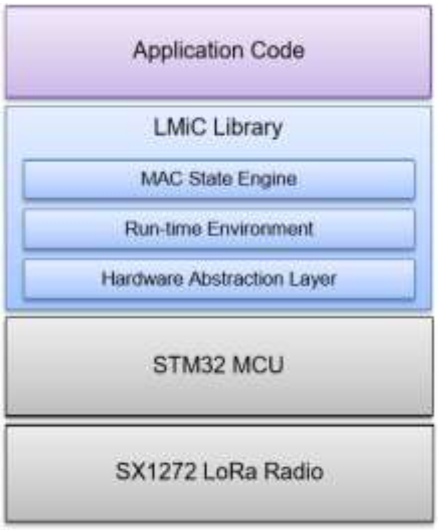


Figure 2. Architecture of LMiC

At the bottom of the architecture, HAL is responsible for communicating with the micro controller unit (MCU) to control the LoRa radio chip. In LMiC, this layer is implemented compatible with the STM32/Cortex-M3 platform. Above HAL, the layer RTE offers a programming model that is based on an event-loop mechanism and manages the tasks of MAC protocol in queues. This layer also takes care of timer to ensure that the constraints of timing of the MAC protocol is satisfied.

The MAC layer, which runs on the RTE, is the implementation of LoRaWan protocol. It has responsibilities of scheduling and perform tasks of listening channels, transmitting and receiving data, and put the radio into sleep state, according to protocol specification. This layer also provides a list of APIs to support the APP layer, which is on top of the architecture and should be implemented by user. The APIs allow user to setup network parameters, set and clear data to send, enable and disable beacon tracking functions, and so on.

In this paper, we introduce a new LoRa MAC protocol, which is implemented based on LMiC library. In our implementation, the RTE and HAL layers are basically maintained. The MAC layer is replaced according to the new design of protocol. Next section will describe protocol design in detail.

3. Protocol Design

3.1. Protocol Overview

In this section we describe the design of protocol with multi-hop communication for linear network in detail. The protocol structure consists of two periods: Network Initialization Period and Data Operation Period.

3.1.1. Network Initialization Period (NIP):

The Network Initialization Period will be performed at the beginning of life time to forming linear network by every node and perform from sink node. The purpose of NIP is to make sure every node in the network is placed at the position that forming a linear network is possible. Besides, NIP is also to make sure that after performing network successfully, every node will go to Data Operation Period at the same time. Because the purpose of protocol is dedicated for the monitoring application in which need coverage long distance and every monitored point is almost placed as line style, so we suppose: Every node is placed at the positions that forming a linear connection is possible and the maximum hop-count of network is pre-known.

3 bits	5 bits	4 bits	4 bits
MSG_TYPE	SENDER_ID	Max_Depth	Passed_Time

Figure 3. Request Initialization Message Format

3 bits	5 bits	1 Byte
MSG_TYPE	SENDER_ID	SRC_FB

Figure 4. Feedback Message Format

To performing initialization network, a Request Init Message (REQ) and Feedback Message (FB) are used. The Figure 3 outlines structure of the REQ. The first byte includes the type of this message and the node ID of REQ's sender. And the byte is to store the maximum depth of network and passed time. The Passed_Time field is to notify receiver that how many time slot already undergone. FB message is illustrated as Figure 4 including message type, sender ID and source of FB which is the node id of the node that successfully upward the first FB message to its previous grade node. Source of FB can be its node id that actively upward FB by itself. If it passively upward FB after receiving FB from its next grade node, the SRC_FB will be same with received FB.

The Figure 5 shows how the network initialize in detail. This period will be performed in certain time interval T_{NIP}. After powering on, every sensor node goes to listening mode to overhear REQ message from its previous grade node. And the sink node start forming linear network by broadcasting REQ message including maximum depth of network and passed time is 1. Every sensor node, after receiving REQ from its previous grade node, it prepares its REQ including Max_Depth information in received REQ and increasing Passed_Time by 1, then send to next grade node. Also, right after receiving REQ, it uses Passed_Time information to calculate remaining time to go to data operating period. Let T_{slot} denote the time slot which is long enough to be greater than time on air of both REQ message and FB message. The remaining time is calculated as follow:

$$t_{remain} = T_{NIP} - Passed\_Time * T_{slot} \quad (1)$$

To make sure that the next grade node received REQ or not, every node (except leaf node) will perform overhearing to track the REQ of next grade node in slot after sending REQ slot. In case of tracking the REQ fail, it increases Passed_Time by 2 and re-send REQ in next slot as last chance.

In this period, before initialization process finish, Sink node will try to receive feedback message which is upwarded from sensor node to be aware of the result which is successful or not based on SRC_FB information in FB. To do so, every node will overhear FB message in short time at the beginning of slot after tracking REQ msg from next grade node successfully. In case of success, leaf node start to send FB in slot after receiving REQ from previous grade node and others node will upward FB to sink node after receiving FB from next grade node. In case of failure, if a node do not track REQ at the second time then it will start send FB in next slot to upward initialization fail status to

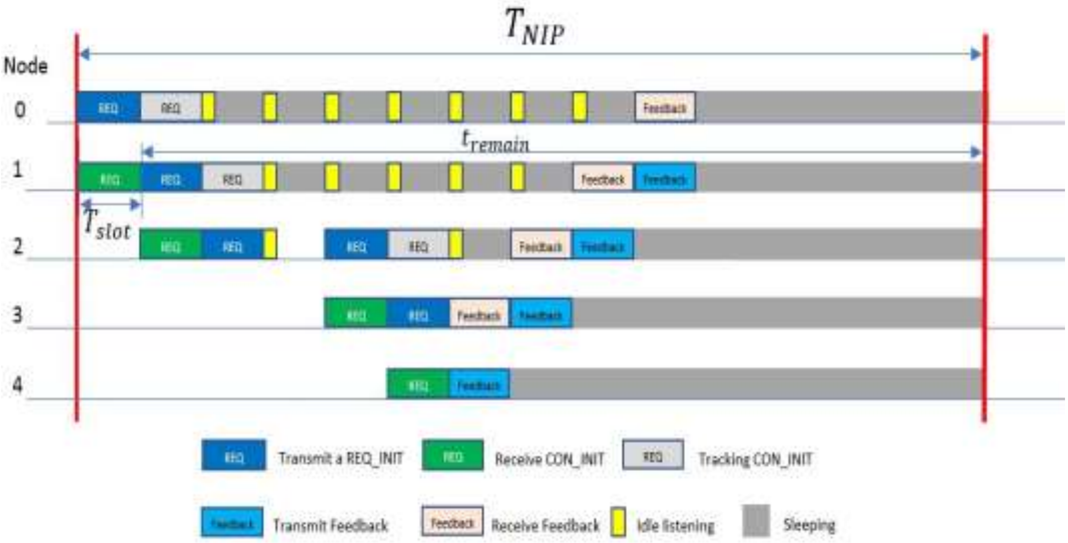


Figure 5. Network Initialization Scheme

the sink node. Besides, even though a node can track a REQ from next grade node, it will schedule event to send

FB once complete REQ reception from its previous grade node to guarantee that FB can always be sent even without receiving FB its next grade node. This event can be scheduled as worst case in which higher grade nodes receive REQ within 2 and is performed at:

$3 * (\text{Max\_Depth} - \text{depth}_n + 1)$ slots after receiving REQ from its previous grade node.

For the required duration time of NIP, we can easily to calculate as worst case:

$$T_{NIP} \geq 4 * \text{Max\_Depth} * T_{slot} \quad (2)$$

3.1.2. Data Operation Period (DOP):

In the data operation period, every sensors node forward data packet in pipeline fashion to send data to sink node. A node can combine the received data packet from next grade node with its own data and send them together. Every node use slot with certain duration time T_{slot} to receiving or sending data packet. The T_{slot} has to be considered greater than duration time for transmitting data packet with maximum length after combining data at node 1. To increasing delivery reliability of the network, we define a mechanism to allow one-time retransmission when first sending time is not successful. The Figure 6 show the data operation schemes of protocol in a cycle. A node, after receiving data packet from next grade node, it will send its data packet in next slot. And right after sending data packet, the node overhears data packet which is sent by previous grade node to be aware that last transmission was successful or not. It will re-send the data packet only one more time after overhearing duration if the tracking is fail. If a node does not receive packet at the first time, it will go to sleep and try to receive after two timeslots. Because every sensor node has one-time retransmission so the nearer node to the sink node has more number of receiving time. Naturally, every node has to send its own data to previous grade node at the last sending slot even if it does not receive any packet from next grade node. With LoRa transceiver, to receiving the packet, the modem will remain RX mode during given period of time to perform receiving packet. If the modem does not detect any packet or reception is fail, it will generate the RxTimeout (RxTO) interrupt and goes back to Standby Mode. In case of successful packet reception, the modem will generate RxDone interrupt at the end of received payload.

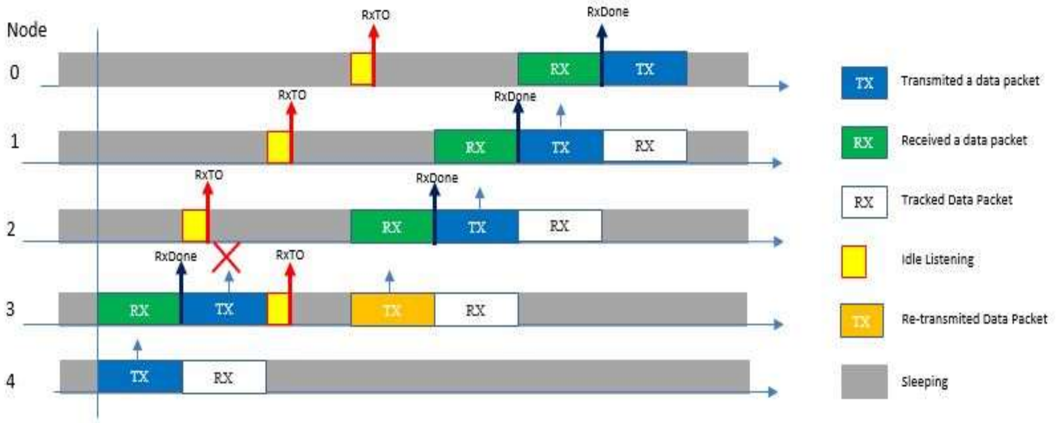


Figure 6. Data Operating Period Scheme with One-Time Retransmission

With the worst case, every sensor node is fail when sending packet at first time and has to retransmit one more time. To avoid collision, the duration of cycle need to be greater the duration to forward data packet from leaf node to sink node in the worst case as follow:

$$dur_{cycle} \geq (3*Max\_Depth + 1) * T_{slot} \quad (3)$$

To evaluate the delivery reliability performance, we will compare one-time retransmission case with no retransmission case. In no retransmission case, every node simply has one time only to transmit data to previous grade node as the Figure 7.

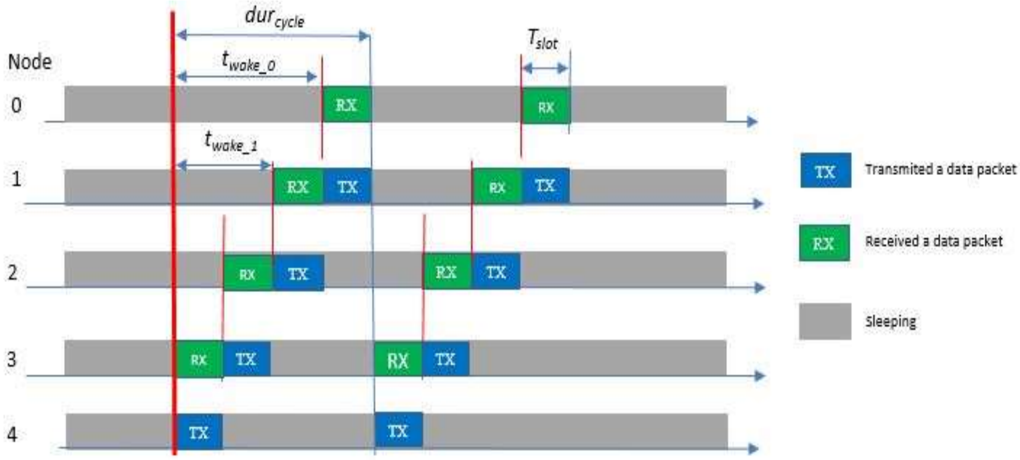


Figure 7. Data Operating Period Scheme Without Retransmission

With no retransmission case, to avoid collision the duration of cycle will be as follow:

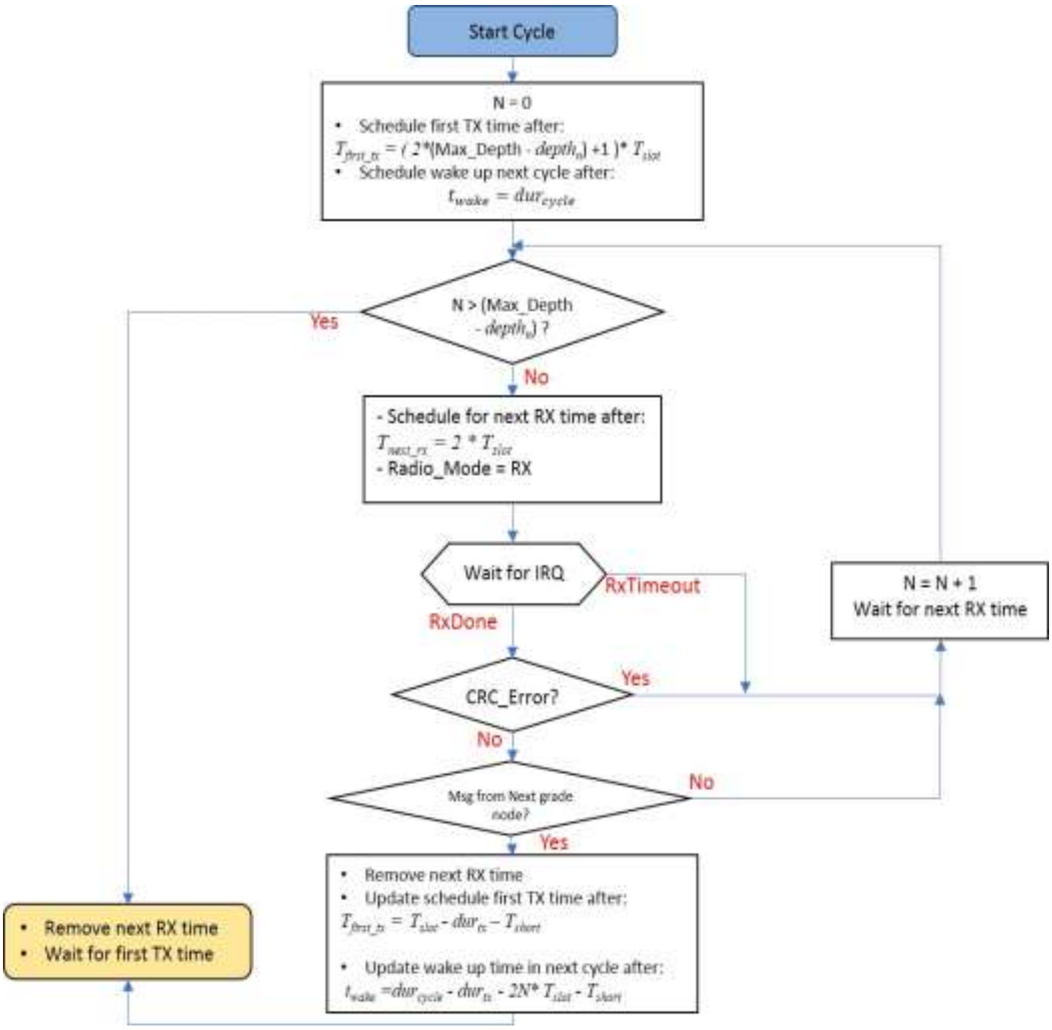
$$dur_{cycle} \geq Max\_Depth * T_{slot} \quad (4)$$

3.2. Wake-up Time Calculation

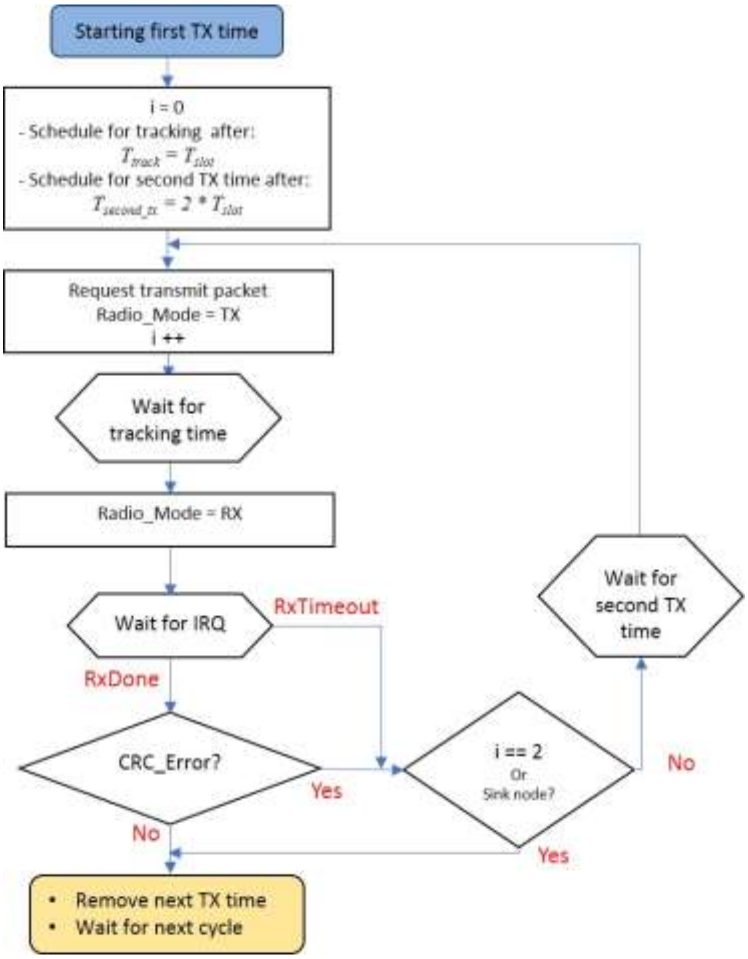
In the first cycle after network initialization period as Figure 7, leaf node will wake up to send data at the starting time of DOP and others node calculate the wake-up time to receive packet from next grade node based on its hop-count (*$depth_n$*) and maximum depth of network as below:

$$t_{wake\_n} = (Max\_Depth - depth_n - 1) * T_{slot} \quad (5)$$

The most important thing to insure protocol operating well is time synchronization from second cycle and so on. Because of clock drift, every node will be loss synchronization after long enough time. So, we provide an algorithm to perform wake-up time calculation during data transmission time. Every node has to wake up to receiving at the same time with next grade node sending. So, a node can be time source of previous grade node. The Figure 8 describes wake-up time calculation algorithm in our protocol. There are two phases: the first phase is for performing receiving data packet from next grade node at the beginning of cycle for each node, thereafter is for performing sending message to previous grade node supporting one-time retransmission. Leaf node will only do second phase and sink node will only first phase. For the intermediate node, they must perform both phase and they will perform first phase until starting first TX time then go to second phase. At the beginning of cycle, every node will schedule wake-up time in next cycle and schedule first TX time for intermediate node. After receiving data packet from next grade node, a node will calculate and update the first TX time and the wake-up time in the next cycle to wake up at the same with the next grade node.



a. RX wake-up Time Calculation Algorithm



b. TX wake-up Time Calculation Algorithm

Figure 8. Wake-up Time Calculation Algorithm

Where:

- dur_{cycle}: duration of cycle
- dur_{tx}: Time on air of packet which is sent by next grade node. It is calculated by formula
- N: Respective receiving times in the cycle
- T_{short}: MCU need a short time to actually change to TX mode in Radio Chip to send packet

3.3. Cycle Duration Requirement and Tradeoffs in LoRa

In this section, we will show several factors affect how we choose the length of preamble and receiving timeout base on synchronization period to avoid loss synchronization. Denote the synchronization period as T_{sync}, and the clock drift rate as r_{clk}. The maximum clock difference [4] between two nodes is:

$$t_{diff} = 2 * T_{sync} * r_{clk} \quad (6)$$

where the factor of two reflects the worst case when each node's clock drifts in the opposite direction.

Between two nodes, the sender has a role as time source. So, we have two worst case: (1) clock of receiver is faster, (2) clock of receiver slower as the Figure 9. Denote minimum time for the modem to acquire lock on preamble as $\boldsymbol{t_{mtone}}$, $\boldsymbol{t_{preamble}}$ is time on air of preamble and $\boldsymbol{t_{timeout}}$ is duration receiver in RX mode to detect valid preamble. $\boldsymbol{t_{mtone}}$ in LoRa is as 4 symbol time which is described in Semtech LoRa transceiver document. The length of preamble and timeout are configured in the register of radio chip. In this protocol, synchronization period is duration of cycle.

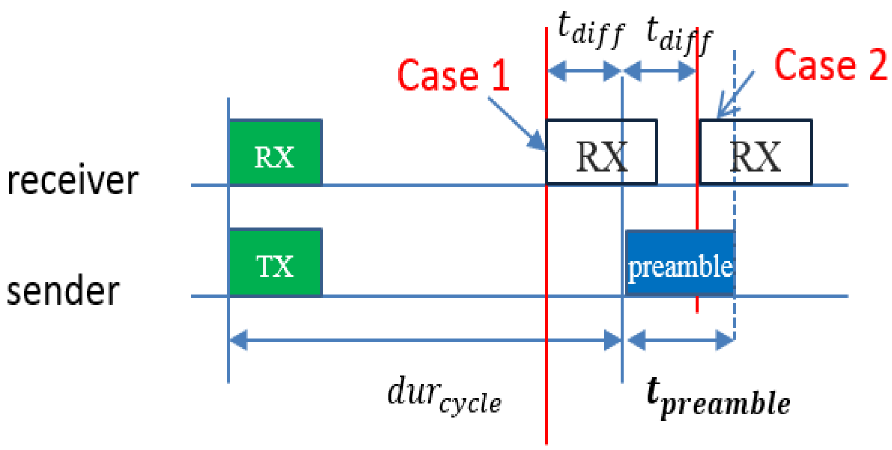


Figure 9. Clock Drift between Two Nodes

In case 1, to detect preamble from sender, the timeout in receiver have long enough to overlap preamble at least 4 symbols. It means:

$$t_{timeout} \geq t_{diff} + t_{mtone} = 2\, dur_{cycle}\, r_{clk} + 4 * T_{symb} \quad (7)$$

In case 2, to detect preamble, the preamble is long enough to overlap with receiving window of receiver at least 4 symbol time, It means:

$$t_{preamble} \geq t_{diff} + t_{mtone} = 2\, dur_{cycle}\, r_{clk} + 4 * T_{symb} \quad (8)$$

In conclusion, to insure that every node does not loss synchronization during data operation, we need to take into account two constraint from Eq. (7), (8) that they have to be satisfied.

4. Performance Evaluation

In this section, performance of the protocol will be evaluated under result of experimentation in real environment. We will perform experimentation in two cases. The first one is for protocol without retransmission. And the second is for protocol with one-time retransmission. They are performed with same configurations, network topology and location. The protocol is deployed on MultiConnect Mdot which is LoRa modules using STM32F411RET Processor and radio chip SX1272 and is developed by MultiTech.

Some important networking parameters and LoRa configurations are shown in Table 1. In this network topology, we deploy five nodes including one sink node and four sensors node. The data message we use in data operation period has format as the Figure 10 including four fields. The first byte stores node id of message and data type of message, which can be temperature or humidity, *etc*. And we use 2 bytes to include sequence of message information. The last field is value of data. A node, after generating is data to send, it has to append data packet of next grade node to send them together. So, the length of data packet can be different between different node. With the maximum depth of

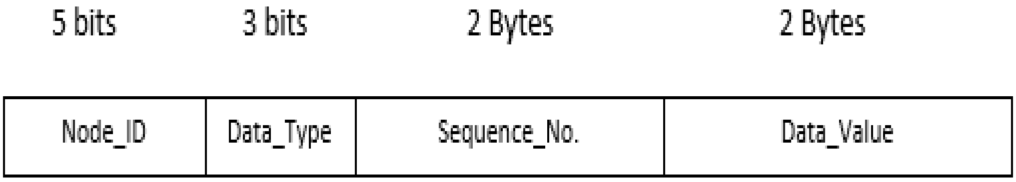


Figure 10. Data Packet Format

network is 4, the maximum length of data packet will be 20 bytes, which happen at node 1.

Table 1. Networking Parameters

Parameters	Value	Parameters	Value
Bandwidth	125 kHz	CRC	Enabled
Transmission power	14 dBm	Frequency	922 MHz
Coding Rate	CR4_5	Preamble Length	8 Symbol
Header Mode	Explicit	RX Timeout	8 Symbol

Table 2. Cycle Duration Parameters

	T_{slot} (ms)	dur_{cycle} (ms)		T_{slot} (ms)	dur_{cycle} (ms)
SF12	1500	6000	SF9	400	1600
SF11	1000	4000	SF8	300	1200
SF10	500	2000	SF7	200	800

Table 2.a: No Retransmission

	T_{slot} (ms)	dur_{cycle} (ms)		T_{slot} (ms)	dur_{cycle} (ms)
SF12	1500	19500	SF9	400	5200
SF11	1000	13000	SF8	300	3900
SF10	500	6500	SF7	200	2600

Table 2.b: One-time Retransmission

In Table 2, we define duration of data packet after calculating the time on air for maximum length as 20 bytes and adding more guard time to avoid conflict among tasks in MCU. The cycle duration is also defined. We choose minimum cycle duration for each case: no retransmission and one-time retransmission.

In experimentation, every node is placed in the campus of university, The Figure 11 shows the position of each node and the distance of between two adjacent nodes. To evaluate the performance of protocol in each case, we take into account delivery

reliability for entire network. Also, a throughput result will be shown to evaluate the capability of protocol. All spreading factor (from SF12 to SF7) will be run one by one. Each test case we allow every sensor node generates 100 data packets to send.

Figure 11. Location Deployment

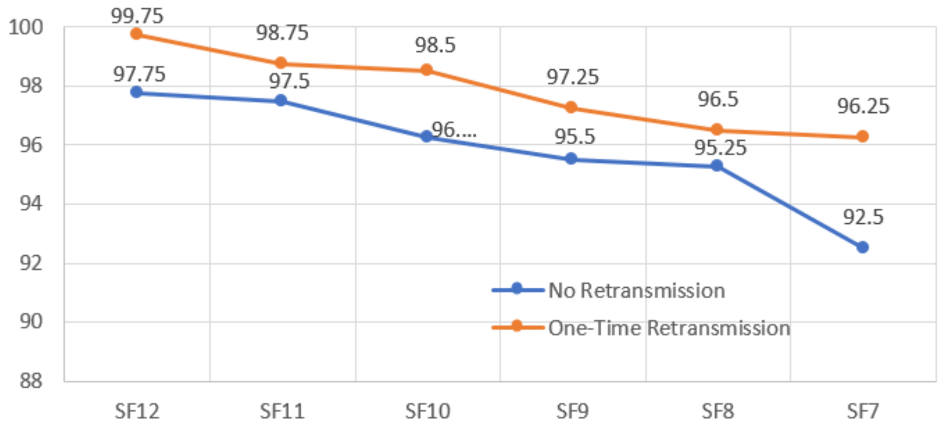


Figure 12. Comparison of Delivery Reliability for Entire Network

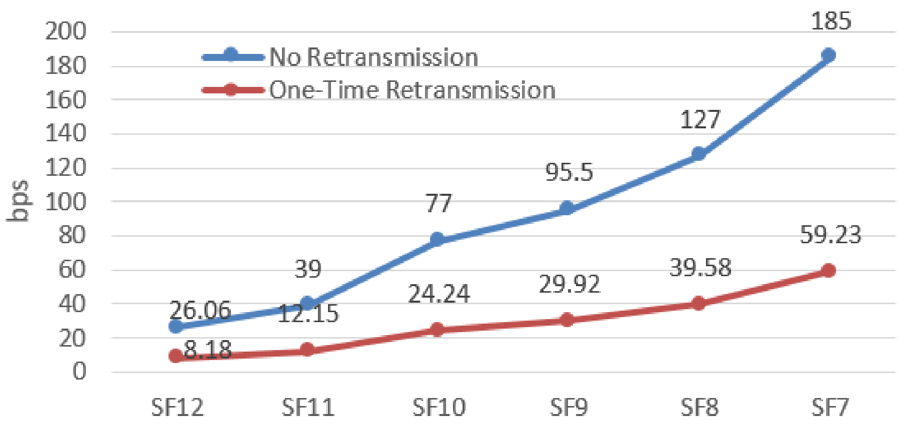


Figure 13. Comparison of Throughput for Entire Network

The Figure 12 and Figure 13 are performance results of entire network. During testing time, number of data packet of entire network is generated 400 packets. The delivery reliabilities of one-time retransmission case for each SF are higher than no retransmission case. However, throughput of network is lower than no retransmission case because every node need to take more time to detect the success of transmission and retransmission.

5. Conclusion

By multi-hop communication, LoRa is potentially very useful for the monitoring system for wide area with long distance. We contributed a protocol with multi-hop transmission for linear network including network construction period and data operating period. Also, we describe a mechanism for time synchronization used in protocol and show the constraints among synchronization time interval, clock drift rate and configurations in LoRa. With the result in performance evaluation part, this protocol is applicable for monitoring application in which need coverage long distance and every monitored point is almost placed as line style such as gas line system, high voltage line system, *etc*.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (KNRF-2017030208).

This paper is a revised and expanded version of a paper entitled "Multi-Hop Linear Network Based on LoRa", presented at the 10th International Conference on Advanced Science and Technology, Ho Chi Minh, Vietnam, February 1-3, 2018 [13].

References

[1] MultiTech Mdot: https://os.mbed.com/platforms/MTS-mDot-F411.
[2] M. Bor, U. Roedig, T. Voigt and J. M. Alonso, "Do LoRa Low-Power Wide- Area Networks Scale?", Proceeding of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, **(2016)**, pp 59-67.
[3] M. Bor, J. Vidler and U. Roedig, "LoRa for the Internet of Things", Proceeding of the 2016 International Conference on Embedded Wireless Systems and Networks, pp. 361-366.
[4] W. Ye, F. Silva and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling", Proceedings of the 4th international conference on Embedded networked sensor systems, pp. 321-334.
[5] Datasheet SX1272: http://www.semtech.com/images/datasheet/sx1272.pdf, Accessed: 24/11/2017.
[6] Semtech Homepage: http://www.semtech.com/.
[7] Ingenu RPMA. http://www.ingenu.com/technology/rpma/. Accessed: 2017-11-27.
[8] Sigfox. http://www.sigfox.com. Accessed: 2017-11-27.
[9] LoRa. https://www.lora-alliance.org/What-Is-LoRa/Technology. Accessed: 2017-11-27.
[10] U. Raza, P. Kulkarni and M. Sooriyabandara. "Low Power Wide Area Networks: A Survey".
[11] LMiC. https://github.com/matthijskooijman/arduino-lmic. Accessed: 2017-11-27.
[12] LMiC. http://www.arduinolibraries.info/libraries/ibm-lmic-framework Accessed: 2017-11-27.
[13] C. Tan Duong and M. Kyung Kim, "Multi-Hop Linear Network Based on LoRa", Advanced Science and Technology Letters, vol. 150, (AST 2018), pp. 29-33.

Author

Duong Cong Tan
Email: duongtan.pfiev126@gmail.com
Address: 44610 Dept. of IT Convergence, Univ. of Ulsan, Daehak-Ro 93, Nam-Gu, Ulsan, Korea